

# **New River Valley Time Bank: Final Project Report**

May 9, 2016

**Project Team:** Kyle Tanous - ktanous@vt.edu  
Aaron Richards - aaronmr2@vt.edu

**Client:** Zoe Belyavsky  
zbelyavsky@blacksburg.gov

**University:** Virginia Tech  
Blacksburg, VA 24061

**Course:** CS 4624  
**Instructor:** Dr. Edward Fox

## Executive Summary

TimeBanks is an international non-profit organization whose goal is to foster strong community relationships and an economy based on home, family, and community instead of money ("Go Deeper | TimeBanks USA" TimeBanks USA). The primary goal of the NRV Time Bank web service is to provide a more usable interface to users and administrators that further promotes TimeBanks' ideals when compared to popular existing time bank management software, especially in regards to inclusivity within the NRV community. The back ends of these established services are well developed and robust. Despite exceptional back ends, the corresponding front end interfaces lack the usability and intuitiveness desired by NRV Time Bank in achieving their stated goals of inclusivity, particularly among its residents lacking general computing experience.

Because relevant functionality problems have already been solved, the primary focus when designing and developing the NRV Time Bank web service is the end-user experience. In particular, the design of NRV Time Bank's web service focuses on how user experience may be improved based on feedback, analysis, and established design guidelines, particularly as outlined in Rex Hartson's *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*.

Using WordPress as a development platform, and established systems as guidelines for functionality requirements, the workload attached to the design and implementation of back-end components is greatly alleviated. This allows for the pursuit of front-end design focused on NRV Time Bank's stated goals of inclusivity. Most low-level functionality is provided by the WordPress platform and available plugins. NRV Time Bank's management system for time bank services distinguishes itself from currently established systems by emphasizing ease-of-use and intuitive navigation, and consequently shortening the learning curve experienced by users.

Alongside being designed with user experience in mind, an important deviation of NRV Time Bank's service when compared to established services is the potential for customization in the future by its administration. Along with providing our service to the NRV Time Bank, long term goals involve providing an accessible template for other time banks. The simplicity and ease of development provided by WordPress combined with a back end developed specifically for time banks will provide both established and future time banks an option for rapid launching of their local organization's web-based services.

As of this report being published, the server image running Ubuntu 14.04 containing the customized WordPress installation with all required plugins, scripts, and interface source material may be obtained on request from the NRV Time Bank staff. Besides the server image itself, documentation, tables, and figures used during design and development are also being made available for reference, alteration, and redistribution.

## Contents

---

• <b>Table of Figures</b> .....	6
• <b>Table of Tables</b> .....	6
• <b>1 Requirements</b> .....	6
○ <b>1.1 Introduction</b> .....	6
○ <b>1.2 Overview</b> .....	6
■ 1.2.1 Objectives.....	6
■ 1.2.2 Scope.....	7
■ 1.2.3 User Roles and Responsibilities.....	8
■ 1.2.4 Replacement of Legacy System.....	8
■ 1.2.5 Production Rollout Considerations.....	9
■ 1.2.6 Terminology.....	10
○ <b>1.3 Functional Requirements</b> .....	11
■ 1.3.1 Functions Required by Users.....	11
■ 1.3.2 Functions Required by Administration.....	12
■ 1.3.3 Security.....	13
■ 1.3.4 Concurrency.....	13
○ <b>1.4 Non-functional Requirements</b> .....	13
■ 1.4.1 Retaining Focus on Usability.....	13
■ 1.4.2 Documentation.....	13
• <b>2 Design</b> .....	15
○ <b>2.1 Introduction</b> .....	15
■ 2.1.1 Objectives.....	15
■ 2.1.2 Scope.....	15
■ 2.1.3 Software Context.....	16
○ <b>2.2 Data Design</b> .....	16
■ 2.2.1 Internal Software Data Structures.....	16
■ 2.2.2 Global Data Structures.....	16
■ 2.2.3 Database Description.....	17
○ <b>2.3 Architectural Design</b> .....	17
○ <b>2.4 User Interface Design</b> .....	18
■ 2.4.1 Overview.....	18
■ 2.4.2 Interface Design Rules.....	18
■ 2.4.3 Components.....	19

- 2.5 Restrictions and Limitations.....20
- 2.6 Testing.....21
- 3 Implementation.....22
  - 3.1 Introduction.....22
  - 3.2 What Will be Implemented.....22
    - 3.2.1 User Account Registration and Login.....22
    - 3.2.2 Posting of User-Created Content.....23
    - 3.2.3 Interaction with User-Created Content.....24
    - 3.2.4 Exchange of Services.....25
  - 3.3 Logistics.....25
    - 3.3.1 Timeline.....25
    - 3.3.2 Hardware.....26
  - 3.4 Version Control.....26
  - 3.5 Documentation.....26
    - 3.5.1 Administration.....26
    - 3.5.2 Users.....27
  - 3.6 Stakeholders.....27
    - 3.6.1 Users.....27
    - 3.6.2 Clients.....27
    - 3.6.3 Administrators.....27
- 4 Prototyping.....28
  - 4.1 Summary.....28
  - 4.2 Functional Prototypes.....28
  - 4.3 Front End Prototypes.....31
  - 4.4 Prototype to Early Implementation.....33
  - 4.5 Conclusion and Moving Forward.....34
- 5 Testing.....35
  - 5.1 Introduction.....35
  - 5.2 Functionality Testing.....35
  - 5.3 Usability Testing.....35
    - 5.3.1 Qualitative Evaluation.....35
    - 5.3.2 Quantitative Evaluation.....36
- 6 Manuals.....37
  - 6.1 User’s Manual.....37
    - 6.1.1 Introduction.....37

- 6.1.2 Account Registration and Login.....37
    - 6.1.3 Posting Content.....37
    - 6.1.4 Viewing and Interacting with Content.....38
    - 6.1.5 Completing Jobs.....38
  - **6.2 Developer’s Manual**.....38
    - 6.2.1 Back-End Structure.....38
    - 6.2.2 Front-End Structure.....39
    - 6.2.3 Migration and Maintenance.....40
- **7 Reflection**.....41
  - **7.1 Timeline**.....41
  - **7.2 Problems**.....41
  - **7.3 Solutions**.....41
- **8 Future Work**.....42
- **9 Acknowledgements**.....43
- **10 Essential Interview Notes**.....44
  - **10.1 Senior Residents of NRV**.....44
  - **10.2 hOurworld User Experiences**.....44
  - **10.3 Early Implementation Impressions**.....45
  - **10.4 Intermediate Implementation Impressions**.....45
- **11 References**.....46

## Table of Figures

---

- **Figure 1.1: Unnecessary steps in navigation (hOurworld).....8**
- **Figure 1.2: Lack of customization options (hOurworld).....9**
- **Figure 2.1: Home screen.....19**
- **Figure 2.2: Request directory.....20**
- **Figure 3.1: Job structure within system.....24**
- **Figure 3.2: Implementation timeline (early March, 2016).....25**
- **Figure 4.1: Early prototype of functional components.....29**
- **Figure 4.2: Service exchange storyboard.....29**
- **Figure 4.3: Service exchange storyboard with focus on navigation.....30**
- **Figure 4.4: Moqups interface prototype.....31**
- **Figure 4.5: System entity relationship diagram.....33**

## Table of Tables

---

- **1.1 WordPress Plugins used by NRV Time Bank Web Service.....7**
- **1.2 User Functionality.....11**
- **1.3 Non-Technical Administrative Functionality.....12**
- **1.4 Technical Administrative Functionality.....12**
- **2.1 Scope of Functionality Classes.....15**
- **2.2 Required Functionality per Component.....17**
- **3.1 Fields Required During Registration.....23**
- **3.2 Content of a User-Created Job.....23**
- **5.1 Quantitative Evaluation Metrics.....36**
- **8.1 Future Goals.....42**

# Section 1

## Requirements

### 1.1 Introduction

---

There are several established software systems currently available (“hOurworld Directory” hOurworld) (“Community Weaver” TimeBanks USA) that act as an interface to the management of a time bank and its services. These established software systems are fully functional, but possess lengthy learning curves when presented to users who may not be familiar with general computer or web-based software use. Because these systems implement the necessary functionality required for time bank management (i.e., functions necessary in facilitating an exchange of services), they serve as useful reference points when developing back-end functionality. The primary focus during development of NRV Time Bank’s service is on improving the front-end interface presented to users, especially with regard to shortening the required learning curve associated with the interface. NRV Time Bank’s web service acts on this focus primarily by promoting intuitive navigation and task completion by reducing required steps where possible and emphasizing recognition and the intended purpose of available actions by providing readily available assistive media (e.g., text instructions for exchange process).

Feedback from users regarding their interaction experiences is able to serve as a continual source of data for the development of future system requirements. Means for reacting to future feedback are provided in the form of administrative access to front-end customization options and a back-end development environment that allows the production of new system components or editing of existing system components (e.g., PHP/JavaScript files and snippets).

### 1.2 Overview

---

#### 1.2.1 Objectives

The primary goal of NRV Time Bank in deploying their new web service is to create a better software system to facilitate the exchange of services relative to established systems. Because established time bank management systems have already implemented and refined required system functionality, the prominent differentiating factor is the quality of the interface presented to users. Enhancing the interface of NRV Time Bank’s service takes the form of improvements in site navigation and the addition of customization options that allow for future reactions to user feedback at both the front and back ends of the system. Both of these characteristics are lacking in current systems (see **Figures 1.1 & 1.2**) and create an unnecessary barrier to entry for new users learning how to use these systems.

Alongside usability goals, our system must record certain user-related metrics (e.g., number of male/female members) in order to comply with requirements defined by the TimeBanks organization to be met by NRV Time Bank.

### 1.2.2 Scope

NRV Time Bank's new software system makes use of an altered version of the standard WordPress database structure to organize standard WordPress data as well as users' account data and created content. All database content resides on the hosting server to serve as the primary data source for keeping track of user-related metrics. Select database content (e.g., Posts submitted by other users) is made available to registered members through a web-based front-end (i.e., a website). The interface to our service is built using a functionally expanded version of the WordPress front-end development environment. Interface development makes use of several available plugins (see **Table 1.1**) that provide additional development tools suited to NRV Time Bank's specific interface implementation needs.

**Table 1.1: WordPress Plugins used by NRV Time Bank Web Service**

- Akismet:  
Included in standard WordPress installation as a comment spam prevention tool.
- Collapse-O-Matic:  
Provides shortcodes that allows easy implementation of expandable/collapsible HTML components. After registering for an account, examples can be seen within the site's request and offer directories (<http://52.32.67.219/index.php/requests/>), where they are used to provide submission forms to users.
- Front-End Users Plugin:  
Creates a separate set of users that are front-end only users, who do not appear in the default WordPress user's area, and allows content to be tailored based on user membership levels.
- Insert PHP:  
Provides short codes for adding PHP code to WordPress posts and pages.
- PHP Code Widget:  
Allows PHP code to be embedded and executed in areas normally not allowed by WordPress. NRV Time Bank's web service leverages this plugin to create sidebar widgets that are capable of generating dynamic content based on database entries.
- Theme My Login:  
Alters appearance of the WordPress login, registration and forgot password pages to be coherent with the site-wide theme.

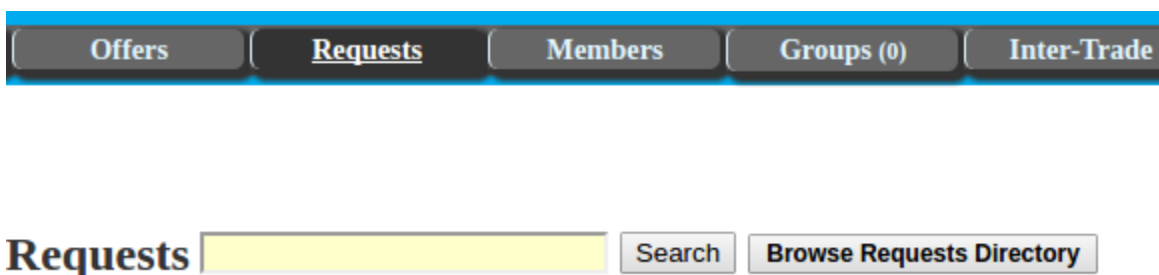
### 1.2.3 User Roles and Responsibilities

Residents of the NRV and NRV Time Bank administrative staff filling non-technical roles exist as the primary demographic of NRV Time Bank’s web service as they represent the majority of the community that NRV Time Bank intends to benefit. NRV residents may employ the new web service to communicate with others and to post job offers and requests that are presented to other users for further interaction. They may post and respond to other users’ posts at any time, with the logistics of the actual job or service being rendered left to the parties involved after using the web service to establish respective parties’ expectations. Administrators in non-technical roles are capable of these same actions, but also possess the capacity to organize special events (e.g., orientations), make announcements that are featured on the site’s home page, and view user-related metrics.

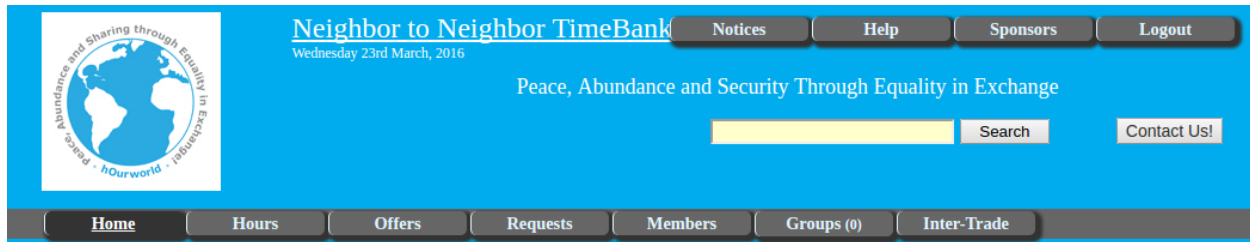
In contrast, NRV Time Bank staff filling technical roles will be effected in majority by back-end design and implementation decisions. NRV Time Bank staff serving in a technical capacity will be responsible for maintaining the service’s back end, fixing instances of software or hardware malfunction, and the future development of the system’s functionality and overlaying interface. Future development includes necessary or desirable (i.e., as deemed by current developers in conjunction with guidance from NRV Time Bank administrative staff) alteration of data structures or their organization within the system. Future development also encompasses refinements to existing functional system components (e.g., scripts that gather and present database contents to users) and potentially the creation of new system components that provide new functionality to users.

### 1.2.4 Replacement of Legacy System

Our system acts as a potential replacement to the popular time bank management systems (“hOurworld Directory” hOurworld) (“Community Weaver” TimeBanks USA) used by the majority of established time banks. While these systems are suitable for the needs of a time bank with regard to functionality, they lack the customization options and high usability desired by NRV Time Bank to assist in working towards their stated goals of inclusion. Figures 1.1 and 1.2 provide examples of these deficiencies.



**Figure 1.1:** hOurworld possesses unnecessary steps in navigation. In this example, clicking on the “Requests” tab leads only to a search bar, even though the demonstrably (based on user feedback, see Appendix 10.2) most common task is viewing and interacting with content directories.



**Figure 1.2:** This theme is not customizable and does not relay the sense of community NRV Time Bank wishes to establish

Our system is similar to these established systems in regards to functionality, but with the addition of increased flexibility in interface customization and a more intuitive interface. These additions are a result of reactions to user and client feedback (see **Appendix 10.1.1 & 10.1.2**).

### **1.2.5 Production Rollout Considerations**

A functional implementation of our system is, as of the publication of this report, available online and is being hosted on an AWS EC2 server instance that can be found at <http://52.32.67.219>. In the future, when funding becomes available, NRV Time Bank's web service should be migrated to a local hosting environment to avoid undesirable regulations, recurring costs, and hardware constraints potentially associated with remotely hosting a web service through AWS. Funding may potentially be obtained through community contributions or outside grants. Besides obtaining the necessary funding, NRV Time Bank staff must ensure they are capable of continuing development and management of their new system without the notable advantages of hosting remotely. In the case of the current EC2 remote host, notable advantages include technical support provided by Amazon, built in version control tools, and specialized physical hosting facilities.

## 1.2.6 Terminology

List of terms used throughout this report, with links to more information given where available.

- **AWS**  
Amazon Web Services; a collection of cloud computing services provided by Amazon.com (<https://aws.amazon.com/>)
- **CMS**  
Content management system. NRV Time Bank's service makes use of the WordPress CMS to manage front-end source code and the EC2 CMS to backup data and configurations belonging to the server instance hosting the web service
- **EC2**  
Elastic Compute Cloud; a virtual machine image that can be used to deploy web-based systems and services (<https://aws.amazon.com/ec2/>)
- **Hours**  
In the context of NRV Time Bank's web service refers to the currency used to compensate other members for work they have performed
- **NRV Time Bank**  
Newly established time bank focused on the New River Valley community, located in Blacksburg, VA (<http://52.32.67.219>)
- **Post**  
In the context of NRV Time Bank's web service refers to a job offer or request created by a registered member
- **SSH**  
Secure Shell protocol; provides a means to remotely access and manage hosting server of NRV Time Bank's web service (<http://www.ssh.com/about>)
- **TimeBanks**  
Parent organization that oversees all established time banks (<http://timebanks.org/>)

## 1.3 Functional Requirements

---

### 1.3.1 Functions Required by Users

Users must be able to perform all tasks required in requesting, performing, and completing an exchange of services. (see **Table 1.2**)

**Table 1.2: User Functionality**

- **Registration and Login**  
Users can register an account and login to view content contributed by administrators and other users. This content includes job posts, announcements, event information, etc. Accounts registered by normal users have their indicated email address verified by NRV Time Bank administrators. Verification functionality is provided by the Front-End Users Plugin.
- **Posting of Content**  
After logging in, users can post requests and offers for jobs. All posts of this type are associated with a specific category (e.g., Lawn Work) for sorting and searching purposes, an optional expiration date, and an optional number of desired workers (this field is only available when creating job requests, not offers).
- **Deletion of Content**  
Users can delete their own previously created posts if they find they no longer wish a post to exist.
- **Responding to Content**  
Users can respond to posts made by others either by sending an email or by making a post in direct response to theirs.
- **Accepting Posts**  
When viewing their own posts, users can view other posts made in direct response to theirs. They can choose a response post to accept, which signifies an agreement to exchange services. It is at this point a job is considered in progress, with logistics being handled by the involved parties from that time onward, with the exception of matters that require intervention by NRV Time Bank administration (e.g., dispute over hours worked).
- **Sending Payments**  
Users can mark a job that is in progress as complete and enter the number of hours worked. The entered value is used to update the amount of hours in possession of the involved parties

### 1.3.2 Functions Required by Administration

Administrators in non-technical roles can perform all the actions granted to users, but with the addition of the following actions that serve to assist in working towards NRV Time Bank's desired sense of community and fulfil the requirements set forth by the TimeBanks organization. (see Table 1.3)

#### Table 1.3: Non-Technical Administrative Functionality

- Posting of Special Content  
Administrators may post the same types of content as normal users, but may also create special posts that can be featured on the home page. These special posts serve to inform normal users about upcoming events or opportunities and to share information deemed important by NRV Time Bank staff.
- Viewing of User-Based Metrics  
Administrators can view user-related metrics based on information given by users during the registration process. These metrics are generated using information such as date-of-birth, gender, and membership type to create comprehensive reports of NRV Time Bank's member activity to be shared with the TimeBanks organization.

Technical staff members do not require the content creation and interaction toolkit granted to users and non-technical administration staff, but instead require functionality that grants them means to access, maintain, and improve front and back-end system components.

#### Table 1.4: Technical Administrative Functionality

- Remote Access to Hosting Server  
NRV Time Bank staff providing technical skills must be able to remotely access the hosting server (typically performed using SSH) to ensure proper operation of the NRV Time Bank web service. Ensuring proper operation consists of routine tasks such as maintenance, which includes updating WordPress, its plugins, and Ubuntu packages when updates become available; and monitoring of server access logs to keep tabs on site traffic and spot potential security risks.
- Access to Development Environments  
Development of the NRV Time Bank web service requires a means to edit and test PHP scripts, JavaScript snippets, and HTML content (regardless of an item's status as a standalone file or embedded within a part of the interface built with WordPress) as a bare minimum. As of publishing this report, all these functions can be performed through terminal-based text editors and the front-end development environment provided by WordPress. These requirements are subject to change as development should continue into the future and new tools potentially become required or desired by developers.

### **1.3.3 Security**

By the very nature of our system, wherein users communicate and will be meeting with each other, there will be security risks. Requests for personal information during registration are kept to a minimum so as to reduce the impact of potential information leaks. Furthermore, the Front-End Users Plugin encrypts all account login data to hinder attempts at obtaining this information. Unregistered users are unable to view content contributed by registered members, but are able to browse special content posted by NRV Time Bank staff. With regard to potential software-level exploits of the web service, WordPress provides software locks to greatly reduce the potential of their occurrence.

### **1.3.4 Concurrency**

Concurrency is a future goal that will bring real time interactivity to our site, namely real-time user-to-user chat and data feeds.

## **1.4 Non-Functional Requirements**

---

### **1.4.1 Retaining Focus on Usability**

The desired outcomes of NRV Time Bank's new web service have been defined by its administrative staff. These desired outcomes consist of increasing inclusivity throughout the NRV community, particularly among its senior residents. NRV Time Bank's web service is based on a design heavily focused on delivering highly intuitive navigation and task completion procedures to its users. The underlying goal necessary in achieving the stated desirable outcomes involves alleviating the cognitive burden associated with learning effective use of an unfamiliar software service. The underlying design of NRV Time Bank's web service incorporates established design guidelines (Hartson, 22.4) along with user and client feedback to create distinct guidelines throughout development. These guidelines serve to ensure progress is consistently being made towards these desired goals and, consequently, desired outcomes. Establishment of the underlying design's focus on usability and as a consistently reliable point of reference helps to ensure deliverable software minimizes the cognitive strain experienced by users learning how to use it.

### **1.4.2 Documentation**

Resources for learning about the TimeBanks organization, its history, and philosophy are made available within the site (<http://52.32.67.219/index.php/about-nrv-timebank/>). This information serves a valuable purpose by providing users with the reasoning behind TimeBanks' ideal of a moneyless economy, as well as providing quantitative evidence demonstrating the success of its implementation across the United States by other time banks.

The broad level overview of the TimeBanks organization and their goals is accompanied by instructional text explaining general use of NRV Time Bank's available functionality. These instructions currently only consist of text on their own dedicated page within the site. The

addition of rich media to these instructions is contingent on obtaining necessary funding and the subsequent migration of the web service to a local server. Future system development can continue significantly less hindered by computing and network resource restraints when separated from potentially dynamic recurring financial costs that NRV Time Bank is currently not adequately prepared to deal with.

## Section 2 Design

### 2.1 Introduction

---

#### 2.1.1 Objectives

Initially, planned design of NRV Time Bank's web service involved creating a wrapper around the mobile API provided by hOurworld ([https://www.hourworld.org/\\_MobileAPI.htm](https://www.hourworld.org/_MobileAPI.htm)) to save us from the task of re-implementing already existing functionality. This initial plan was discarded based on lacking documentation and undesirable restrictions the API would place on possible functionality. Front-end flexibility is paramount in delivering a software system that provides a more usable interface when compared to established interfaces to time bank services ("hOurworld Directory" hOurworld) ("Community Weaver" TimeBanks USA). Abandoning the potential convenience of using an existing API is necessary to ensure delivered software maintains the objective of shortening the learning curve presented to users.

#### 2.1.2 Scope

The scope of development is mostly confined to the provision of front-end functionality for both users and administrators. (see **Table 2.1**)

#### **Table 2.1: Scope of Functionality Classes**

- **Essential Functionality:**  
Core time bank functionality requires that users be able to request, perform, and complete the exchange of services using NRV Time Bank's web-based interface. This includes posting offers and requests, responding to offers and requests, and communicating with other users. Offers, requests, and user information must be recorded in order to report user-based metrics on a monthly basis.  
Administrators must be able to communicate with users in a way that differentiates them from normal user communications. Administrators must be able to create obviously significant posts to inform users of upcoming events and opportunities, such as orientations or volunteer work available within the organization.
- **Desirable Functionality:**  
Secondary goals involved in setting up an exchange of services should be built into NRV Time Bank's web service. This class of functionality encompasses non-necessary information delivery, such as estimated driving distance; and non-common tasks, such as requesting multiple volunteers for a single job.

- **Future Functionality:**  
NRV Time Bank is actively working to procure the necessary funding to setup their own local hosting environment. The resulting reduction of computing and network resource restraints will allow development to continue into superfluous features, such as real-time data and real-time user-to-user chat.

### **2.1.3 Software Context**

The interface provided to users by NRV Time Bank's system aims to provide a relatively more intuitive way to carry out tasks associated with intrinsic time bank functionality. It will be similar to interfaces provided by established systems with regard to actions available to users, but with the advantage of increased usability. Increased usability results from the prioritization placed on maintaining high learnability in the majority of design materials. Additional benefits of designing the system free from external dependencies stem from the ongoing control over all aspects of the system maintained by NRV Time Bank. This presents opportunities for improvements to the system to continue as user feedback becomes available. This is important to the longevity of the service and increasing its potential positive impact on not only the NRV community, but also communities associated with other established time banks across the United States that may decide to adopt the system.

## **2.2 Data Design**

---

### **2.2.1 Internal Software Data Structures**

User data objects contain a user's name, email, amount of hours, and a unique ID that allows queries for user data objects to remain static despite user information's potential to change.

User account information is handled by the Front End Users Plugin's (FEUP) CMS. Because certain information required about users is subject to change, (based on what must be reported to TimeBanks) additional user data is stored separately as user field objects. This data consists of information entered by users upon account registration, such as date of birth and gender.

User post objects contain a reference to the user who posted, a category, a description of the task to be completed, an optional expiration date, and optional number of requested volunteers. Classification of posts as offers or requests is determined by the WordPress page used for submission and its associated database insertion script.

### **2.2.2 Global Data Structures**

User data and posts are stored in tables within the MySQL database provided by the standard WordPress installation. These tables are written to by standalone PHP scripts residing on the hosting server, and read by PHP/JavaScript embedded within WordPress pages. PHP script location is determined by level of required access to the database. Scripts which contain

administrative login information for the MySQL service are isolated as individual files to emphasize differences in permissions and capabilities between scripts.

### 2.2.3 Database Description

The MySQL database responsible for storing user data and posts contains five tables beyond those created during WordPress installation that were created during development or provided by the FEUP. These additional tables are user data, user fields, requests, offers, and metrics. The user data and fields tables contains user data objects and associated information provided during account registration. These tables serve to associate posts with their owners, and to generate user-related metrics required by TimeBanks. The requests and offers tables contain user post objects and are accessed upon request by users of the service. The metrics table holds values to be reported to TimeBanks and is updated as jobs are created, interacted with, and completed. Values belonging to the metrics table cannot be generated based on normally accessible user data or fields, so they are stored separately in their own table.

## 2.3 Architectural Design

---

Currently available systems provide the necessary functionality required by a time bank, and are recreated in NRV Time Bank's system. (see **Table 2.2**)

**Table 2.2: Required Functionality per Component**

- Accounts
  - Users and administrators must be able to register and log into an account
  - Administrators must verify user accounts as part of the registration process
- User-Created Content
  - Users can create and post job offers and requests
  - User provides category, description of job, optional expiration date, and optional number of requested workers per post
  - Posts are associated with user IDs to provide access to contact information
- Administrator-Created Content
  - Administrators can create special content besides offers and requests. This special content may be announcements, information about upcoming events, etc.
  - Content created by administrators should be accessible to all users, not only registered members
  - Users can contact the owner of a job post via email
- Support
  - Contact information of time bank staff must be available
  - Administration must be able to settle disputes (e.g. disagreement on hours worked)

The WordPress database contains the tables for user data and posts, which are read by PHP scripts embedded within WordPress pages, and written to by standalone PHP scripts residing on the host server. The embedded scripts that read from the database perform queries based on user-related and passed parameters and displays the results in formatted row entries to be viewed by registered members. Standalone scripts possess elevated permissions that allow the execution of insertion, deletion, and alteration queries.

## **2.4 User Interface Design**

---

### **2.4.1 Overview**

Front-end content is encoded in HTML5 and CSS. All pages of the website interface possess a common theme with a static top bar menu accessible from any page within the site. The majority of HTML content displayed by WordPress pages is generated by PHP scripts embedded in WordPress pages, and formatted to adhere to the overarching interface theme despite the content type. This helps to emphasize actions and promote recognition of tasks. This is of particular benefit to users who are less experienced with using web-based services.

Information obtained and presented by embedded scripts can be searched and sorted based on user-defined filters, but default to the most common options to reduce average necessary steps in finding relevant posts.

Educational and instructional media is accessible from the top bar menu present on all pages. This information exists to assist users in navigating the service and performing basic tasks required during the process of exchanging services.

### **2.4.2 Interface Design Rules**

The overall aesthetic theme must maintain coherence across all pages, with similar organization of actions and data between pages to promote recognition of actions available to a user, despite their location within the site.

More common tasks should require fewer actions to complete. This idea is paramount to creating an easily learned service. Implementation of this rule requires maintaining groupings of common tasks to promote both recognition and remembrance of available actions.

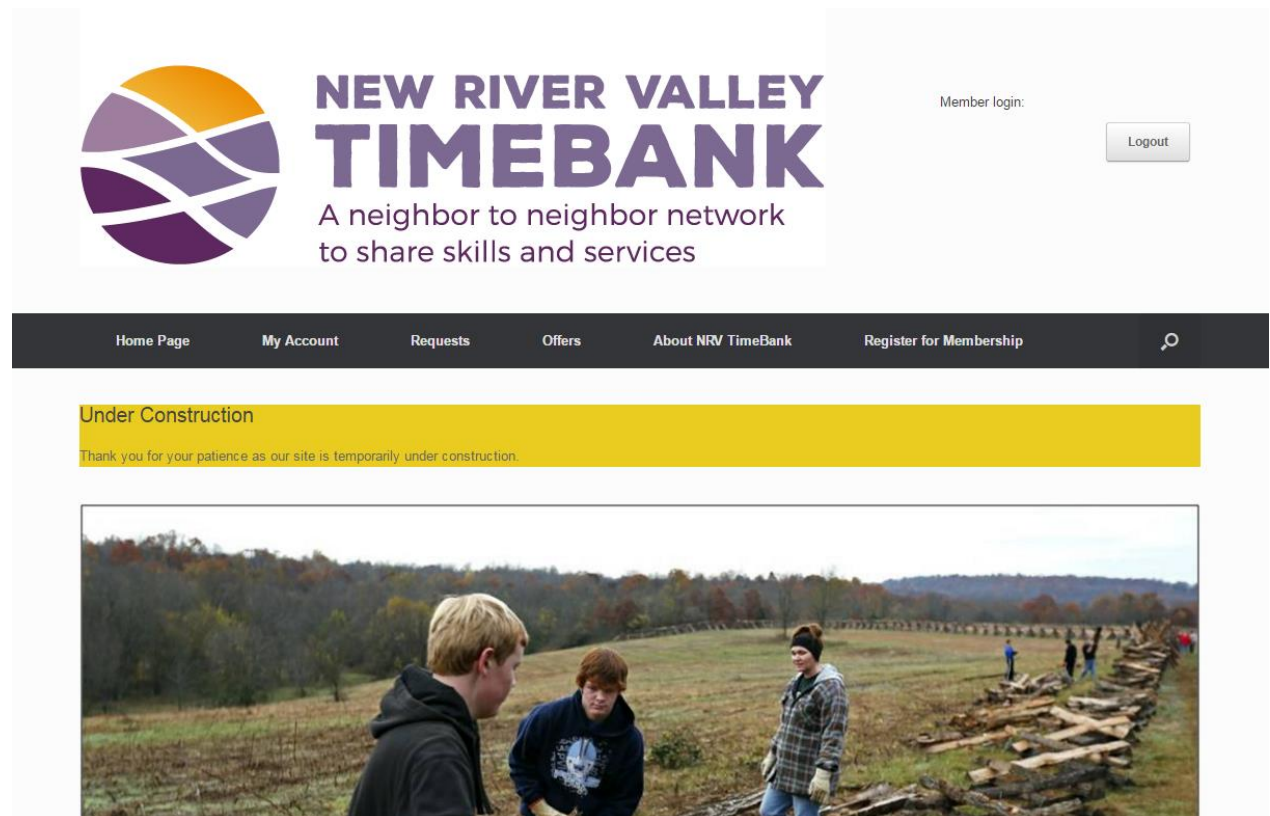
Back button navigation must be intuitive. As an example, if a user changes the sorting filter for a list of posts, the back button should not take them to the previous sorting filter. Instead, they should be redirected to the most recently accessed page before navigating to the page containing the list of posts occurred.

Menu items should be accessible without switching pages if possible. This serves to reduce potential confusion in navigation and unnecessary switching between pages.

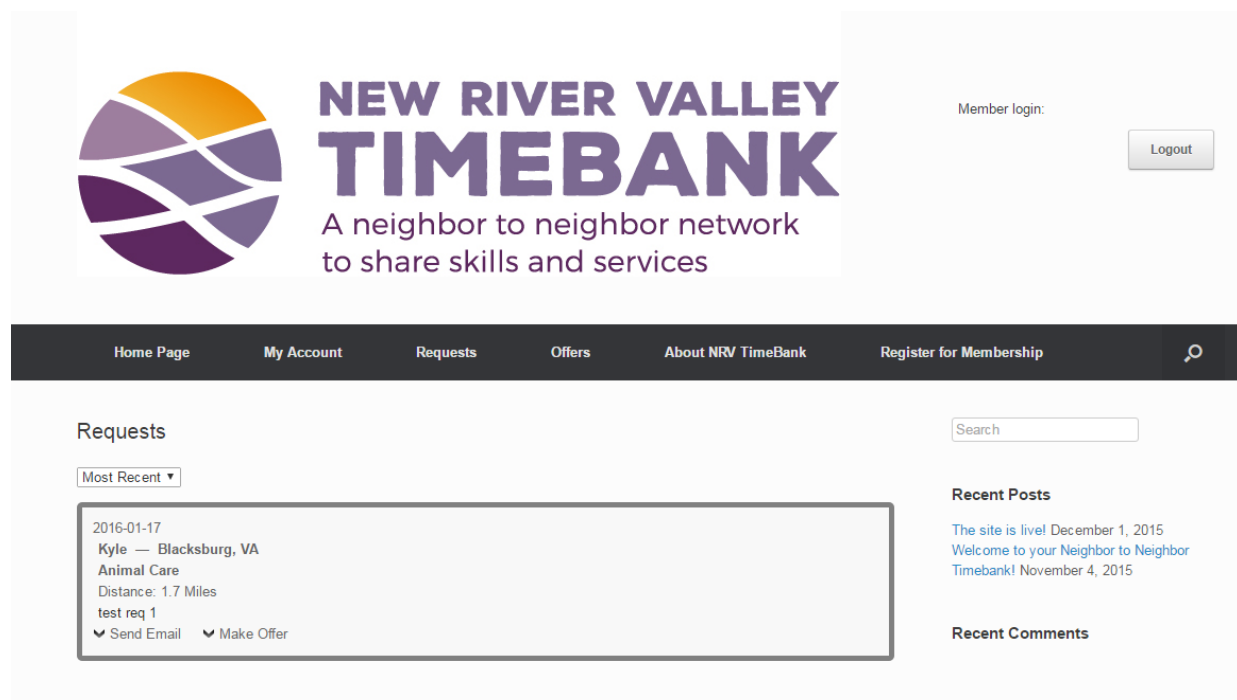
### 2.4.3 Components

The top bar menu is present across all pages and contains links to pages corresponding to more commonly performed tasks. Menu items located here are expandable on mouse over as well as clickable.

The homepage of the website interface acts as a hub for all users. Unregistered members are presented the option of registering for membership and links to pages containing further information about TimeBanks. Registered members are able to easily navigate to post directories, post submission pages, and their account page (see **Figures 2.1 & 2.2**) to take advantage of the functionality provided by NRV Time Bank's software system.



**Figure 2.1:** Home screen of service that can be found at <http://52.32.67.219>



**Figure 2.2:** Request directory of service, accessible to registered members at <http://52.32.67.219/index.php/requests>

Submission forms are used in the performance of tasks associated with most common functionality provided by the website’s underlying system. Submission forms are designed with obvious explanations regarding each field and predefined lists for certain field entries to assist in maintenance of consistency across user-generated content.

## 2.5 Restrictions and Limitations

Aspects of our design that require high computational power are not feasible to implement due to hosting on an AWS EC2 instance and the associated potential costs. This class of design ideas relies on higher computing power than can be utilized using EC2. These ideas namely consist of providing real-time data feeds and chat functionality to users. Upon obtaining the necessary funding, NRV Time Bank will have the option of migrating the service to a local server and begin implementing these more resource-intensive features. The NRV Time Bank is on a soft deadline for release of the software service to the NRV community. Convenience-focused functionality will be treated as a secondary implementation concern to accommodate the current release timeline and allow more effort to be applied to implementing necessary functionality and improving the interface to NRV Time Bank’s services.

## 2.6 Testing

---

Testing of NRV Time Bank's software system is carried out manually, as testing scenarios are short (typically less than 2 minutes per test), and can be performed remotely with sufficient efficiency so long as involved testers maintain reasonable levels of contact, which is easily achieved by using text, voice, or video chat software.

Testing generally incorporates three phases per implemented function. The process begins with establishing correctness of a function by examining inputs and checking for proper output. Because proper output generally consists of database alterations, it is highly important to ensure there are no unintended effects resulting from a query. This requires populating the database with varying groups of data to account for any unexpected scenarios and experimenting with the timing and order of operations to test data locks.

After establishing functional correctness, efficiency is considered. During this phase the source code of the implemented function is optimized and steps are taken to reduce unnecessary use of hardware resources. This is of particularly high priority due to the unwanted potential costs associated with using excessive hardware resources of the current EC2 service host.

Finally, we take steps to ensure any information presented to the user maintains coherence with the design of the overarching website interface. This portion of testing requires feedback and reference of previously stated guidelines to ensure data is being presented as thoroughly and as legibly as possible to users.

## **Section 3 Implementation**

### **3.1 Introduction**

---

Implementation of NRV Time Bank's web service follows a cycle of design, development, and testing for each group of functions to be implemented. This primary cycle is followed by iterative improvements applied to the cumulative result. Groups of implemented functions are represented as groups of WordPress pages that all provide similar functionality to users. Stakeholders of the system consist of citizens of the NRV, who will be interacting with the front-end interface presented by WordPress; the NRV Time Bank staff, who will be responsible for maintaining the site's content; and the site administrators, who will be ensuring the site is up to date and providing technical support to users.

The system will not be related to currently available systems aside from providing similar functionality that is inherent to all time banks. Web-based API's are available through established time bank software systems, but the idea was abandoned in favor of the benefits provided by designing and developing the system from the ground up. The most notable benefit of this approach is the significantly higher levels of control that can be maintained by NRV Time Bank. The primary drawback of this approach is the increased stress applied to an already unaccommodating release timeline, but is far outweighed by benefits resulting from retaining of complete control and the shedding of external software dependencies.

### **3.2 What Will Be Implemented**

---

The front-end of our system will be provided to users by a customized WordPress deployment, where pages consist of HTML5 generated by PHP and JavaScript. The back end consists of a MySQL database that is accessed via embedded and standalone PHP scripts. Through proper implementation, the end result is a simple code base that can be easily maintained and improved by future administrators of the site. Required functionality to be implemented is broken into the following groups.

#### **3.2.1 User Account Registration and Login**

Users must be able to register and log into an account before they can view the site material contributed by registered members. Accounts must be verified, to prevent unmanned account creation. Implementation of these features will be handled by the FEUP, which is designed specifically for the purpose of managing front-end user accounts. It provides the added benefit of easy content restriction based on user membership type.

Account registration consists of the following required fields to be used for login and metric gathering. (see **Table 3.1**)

**Table 3.1: Fields Required During Registration**

- Contact Information
  - Email - basic contact information
  - Phone number - basic contact information
- Demographic Information
  - Date of birth - used in keeping record of junior and senior members
  - Gender - used in keeping record of gender demographics
- User information
  - Name - shown alongside job post (only displayed to verified, logged in members)
- Information for Functionality
  - Address - used in estimating driving distance, never shown to other users until voluntarily revealed by parties involved in an exchange
- Login Information
  - Password

**3.2.2 Posting of User-Created Content**

After logging into their account, users will be able to create job offers and requests that will be stored in their respective tables residing in the WordPress database. The interface to offer and request creation is provided by WordPress pages containing HTML submission forms which take several user-defined inputs that classify the post. After classification, post information is passed to a PHP script with the necessary permissions to perform the required insertion query.

Offer and requests posts consist of the following items. (see **Table 3.2**)

**Table 3.2: Content of a User-Created Job**

A user-created post is represented as a database entry containing the following information:

- Post ID - unique identifier for each post made
- Poster ID - to associate with original poster
- Body - details of job
- Posting and expiration dates - allows posts to have a user-defined lifespan
- Associated response ID - only applicable to offers and requests that have been made in direct response to another user's post or are part of an active exchange
- Category - selected by users from a premade list to assist in organization of content
- Number of volunteers - used in the instance somebody wants to request multiple workers (request only)

### 3.2.3 Interaction with User-Created Content

After logging into their account, users will be able to respond to posts made by others (i.e. posting an offer in response to a request and vice versa). The original poster of an offer or request can view these responses and choose to accept one. The original post becomes a job in progress after a response is accepted. The interface to offer and request interaction is provided by expandable forms generated for each post presented on a given WordPress page. Directory pages display all open jobs. A job is considered open if a response has not yet been accepted by the original poster. The directory pages also contain expandable forms for messaging post owners and submitting responses. A user's own open jobs can be viewed on their personal account page.

A job in progress consists of a request and one or more offers. These are associated by pointers to each other within the job post data types stored in the WordPress database. After completion, all posts associated with the job are deleted, and hours of involved parties are altered to reflect the number of hours worked.

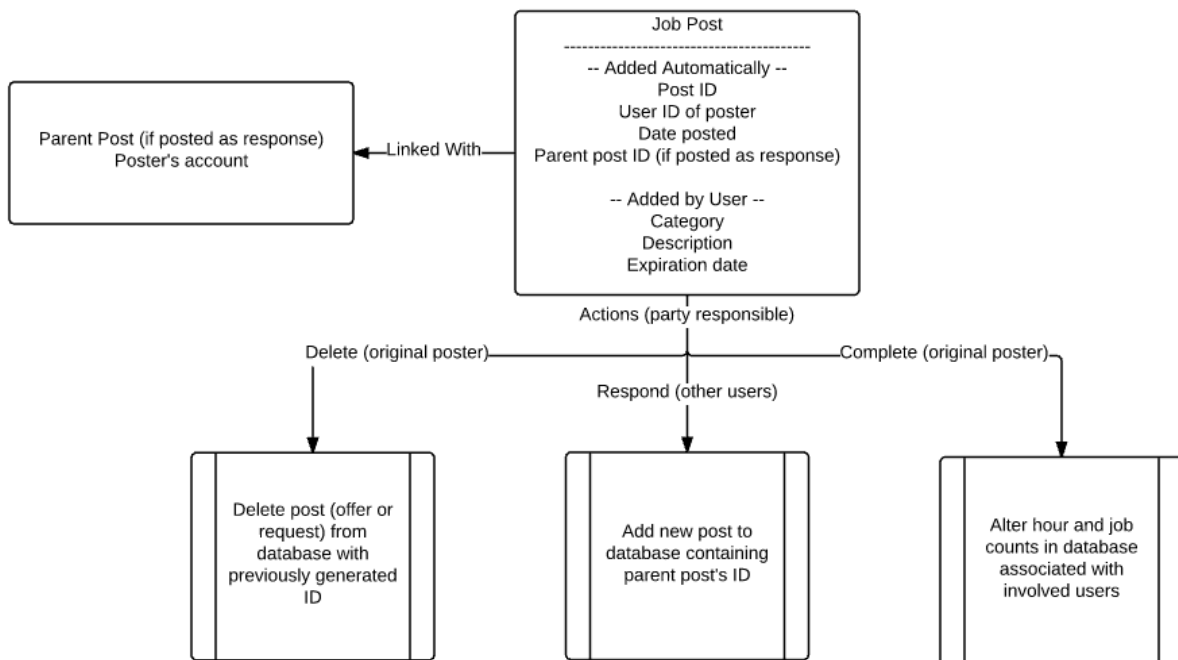


Figure 3.1: Job structure within system

### 3.2.4 Exchange of Services

After logging into their account, users will be able to accept job postings made by other users in direct response to their own. It is at this point that a job is considered in progress, the logistics of which are then shouldered by involved parties. After a job is complete, hours spent performing the job are reported via a form available through the requesting user's account page. In the event that there is a dispute between users over the amount of hours worked, a member of NRV Time Bank's staff will be allowed to intervene and act according to their best judgement.

The linking of an offer and a request is represented as pointers to each other's IDs stored in the posts' respective database table entries. Job completion is carried out by the requester of a job. On completion form submission, the amount of hours and completed jobs associated with the involved users are updated via PHP scripts capable of altering the MySQL database.

## 3.3 Logistics

### 3.3.1 Timeline

The WordPress deployment is the initial priority, as all our functionality is built upon the WordPress framework. Subsequently, the ability to register and log into an account will be implemented. Once accounts are able to be created we will implement necessary functionality to allow users to generate content. Finally, we will implement functionality to provide users with the ability to interact with the content of others. Each phase of implementation will undergo testing as development finishes to ensure we create organized source code that can be easily managed.

Task #	Task	% Complete	Status	Day Started	Planned Completion	Actual Completion	Task Prerequisites	Priority	Notes
1	Setup AWS EC2 instance	100	Completed	1/18/2016	1/20/2016	1/18/2016		High	
2	Install and setup standard Wordpress deployment	100	Completed	1/18/2016	1/20/2016	1/19/2016	1	High	
2.1	Customize theme	100	Ongoing	1/19/2016	1/20/2016	1/20/2016	2	Medium	Improved periodically based on user and client feedback
2.2	Install and setup necessary Wordpress plugins	100	Completed	1/20/2016	1/20/2016	1/20/2016		High	
3	Setup MySQL database		Ongoing	1/20/2016	5/1/2016		2	High	Ongoing process; changes always possible
3.1	Implement tables to store offer and request structures	100	Completed	1/20/2016	1/21/2016	1/21/2016	3	High	
3.2	Alter plugin tables to accommodate our usage	100	Completed	1/20/2016	1/25/2016	1/26/2016	3	High	
4	Add placeholder pages for user interface to functions	100	Completed	1/20/2016	1/20/2016	1/20/2016	3	Low	
5	Add sidebar widgets to homepage	95	Postponed	1/21/2016	1/28/2016		4	Medium	
5.1	Login widget	100	Completed	1/21/2016	1/22/2016	1/22/2016			Provided by Plugin
5.2	Register link	100	Completed	1/22/2016	1/22/2016	1/22/2016			Provided by Plugin
5.3	Recent posts feed	90	Postponed	1/26/2016	2/1/2016		3.1		Postponed until we determine what information should be visible
6	Implement account management pages	100	Completed	1/27/2016	1/28/2016	2/5/2016	3	High	
6.1	Account overview page	100	Completed	1/27/2016	1/28/2016	1/28/2016			
6.2	My jobs page	100	Completed	2/2/2016	2/3/2016	2/5/2016			
6.2.1	Post deletion	100	Completed	2/2/2016	2/3/2016	2/5/2016			
6.2.2	Accepting responses	100	Completed	2/2/2016	2/3/2016	2/5/2016			
7	Implement directory pages	75	On Schedule	2/8/2016	3/15/2016		3	High	
7.1	User to user messaging	99	Completed	2/9/2016	2/9/2016	2/9/2016			Ran into issues with email server
7.2	Response submissions	100	Completed	2/8/2016	2/9/2016	2/9/2016			
7.3	Sorting	100	Completed	2/12/2016	2/18/2016	2/19/2016			
7.4	Display driving distance	90	On Schedule	2/22/2016	3/5/2016				Google Maps web API
7.5	Search	0	Not Started						Search by category and key words
8	Implement job creation pages	60	On Schedule	2/10/2016	3/15/2016	2/28/2016	3	High	
8.1	Category select	100	Completed	2/15/2016	2/18/2016	2/17/2016			Categories provided by NRV Time Bank
8.2	Expiration date	75	On Schedule	2/20/2016	3/7/2016				Need to determine best way to automatically delete expired posts
8.3	Request multiple volunteers	20	On Schedule	2/25/2016	3/15/2016				How to associate single request with multiple offers
9	Metric gathering	80	On Schedule	1/28/2016	5/1/2016		3	High	Required by TimeBanks organization
9.1	User related metrics	100	Completed	1/28/2016	2/5/2016	2/24/2016			
9.2	Sponsor related metrics	50	On Schedule	2/25/2016	3/12/2016		10		Need to implement sponsor related functions
10	Implement donation options	50	On Schedule	2/28/2016	3/25/2016		2.2	Medium	
10.1	Add different membership levels	100	Completed	2/28/2016	2/28/2016	2/28/2016			Provided by plugin
10.2	Add Paypal support	50	On Schedule	2/28/2016	3/10/2016		2.2		Maybe credit card support?
11	Instant messaging options	0	Not Started					Low	Stretch goal
12	Real time offer and request feeds	0	Not Started					Low	Stretch goal

Figure 3.2: Implementation Timeline as of early March 2016

[https://docs.google.com/spreadsheets/d/1SHLPf81\\_jwM11KIhp8I81wVyNjXbmjQ2RIyavTje\\_yc/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1SHLPf81_jwM11KIhp8I81wVyNjXbmjQ2RIyavTje_yc/edit?usp=sharing)

### **3.3.2 Hardware**

Currently, we are making use of hardware resources provided by Amazon's EC2 service. The server instance runs Ubuntu 14.04 with relatively limited resources in terms of storage (16 GB) and computing on a T2 micro tier instance type. The only hardware required for administrative purposes is a machine with the capability to connect to our EC2 instance via SSH.

Users can access the service through a web browser. Because we are only targeting the New River Valley area, storage and computing requirements are not unreasonably large, and in the short term can be accommodated by our currently limited hardware resources. As of publication of this report the EC2 server can be maintained free of charge provided we keep usage of hardware resources to a minimum.

## **3.4 Version Control**

---

Most of our project's version control is handled by our EC2 and WordPress's respective CMSs. The EC2 service provides version control for our server's configurations and local data. WordPress provides version control for the source code of all pages within the site, as well as local database contents. Through EC2, we are able to create backup images of our entire EC2 server instance in case mistakes are made. This allows us to save local data (user accounts, job postings, etc.), server configurations (port and security settings), and application configurations (MySQL, WordPress). These backup images also prove useful in the event the service must be relocated to a new hosting environment. The WordPress CMS maintains a revision history of all pages within the site, which can be used as restore points in case bad edits are made to the source code of a webpage.

## **3.5 Documentation**

---

### **3.5.1 Administration**

Administrative documentation covering details of operating and maintaining NRV Time Bank's system will be provided to members of its staff. This will cover matters specific to our system, such as manually altering database contents, registering users, and troubleshooting WordPress related issues through the provided dashboard. This area of documentation will be written for those familiar with Linux and server management in general in order to accommodate the release timeline of the NRV Time Bank service. Administration requires at least a moderate understanding of Linux, some of its commonly used applications, and how to use them effectively. Given our time constraints, it is not be feasible for NRV Time Bank to provide their own documentation for Linux and applications relevant to their system. Instead, compiled resource lists can be provided to those who are unfamiliar with these aspects of server management.

### **3.5.2 Users**

Documentation outlining functionality provided by NRV Time Bank's software service and instructions for use are available through the system's interface and can be found at <http://52.32.67.219/index.php/about-nrv-timebank/>. Only text will be made available upon initial release of the service. Rich media will be included in instructional content after local hosting facilities become available to NRV Time Bank and restraints imposed by the current EC2 host can be abandoned.

The planned video portion of this documentation will consist of screen recordings of task procedures involved in the service exchange process. Images and text will accompany each other. Images will consist of screen captures relevant to performing service exchanges. Text will accompany images to thoroughly describe intended procedure and actions available to users at various points throughout the service exchange process. The text and image portions of the documentation will provide more thorough coverage of procedures for those that care to learn about it. Video guides will provide relatively shallow coverage, but should suffice for typical use cases.

## **3.6 Stakeholders**

---

### **3.6.1 Users**

The primary benefit NRV Time Bank is offering with their web service is an increase in usability. Because of this, users and their feedback are the driving forces behind the majority of design decisions made. Feedback from senior citizens of the community will be of the highest priority to assist NRV Time Bank administration's goals of their inclusion within the NRV community.

### **3.6.2 Clients**

NRV Time Bank staff know what needs to be done to ensure their new system conforms to the standards set by the TimeBanks organization. Their feedback and guidance are heavily focused on metrics and requirements gathering. The remaining minority of client feedback consist of suggestions for additional functionality that exist outside of the normal scope of time bank software systems.

### **3.6.3 Administrators**

System administrators will be most impacted by implementation decisions, and as such possess the most meaningful feedback during implementation. Administrators of the system are responsible for manual database alterations, settling disputes between users, tech support, server maintenance, and system maintenance. Their feedback will help us ensure that our back-end is simple in structure, easily maintained, and easily built upon.

## Section 4 Prototyping

### 4.1 Summary

---

Design models developed during information gathering and storyboarding (see **Appendix 10.1.4: Storyboards & 10.2: Interview Samplings**) proved extremely useful in the development of prototypes. In addition, the solid foundation of requirements developed during information gathering and design allowed the efficient creation of prototypes. These requirements were frequently referenced during prototyping and led us to pursuing a prototype model primarily focused on the front end of our service.

Despite the focus on usability, front-end prototypes were supported by rudimentary functionality prototypes (see **Figure 4.1**). These early functionality-focused prototypes provided a foundation and guidelines for the design and development of front end prototypes and subsequent interface implementation.

The prototypes conformed to the vision for the system constructed based on analysis, design, and feedback from users and clients. These prototypes allowed thorough exploration into potential front-end options without being hindered by the logistics of back-end functionality. This section documents the prototyping process and how it is linked to previous information gathering, design, and subsequent implementation.

### 4.2 Functional Prototypes

---

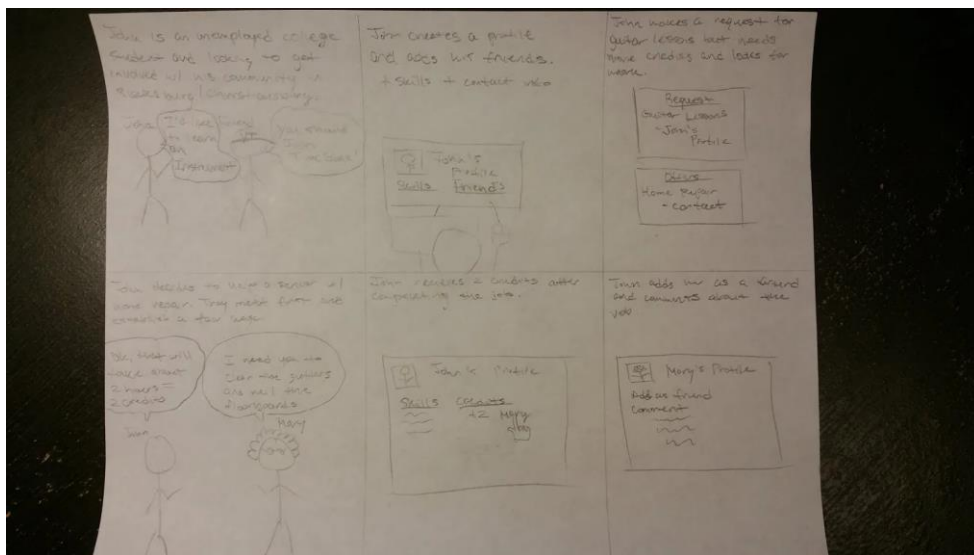
Early prototypes of NRV Time Bank's system defined basic capabilities of functional system components, but were still tied heavily to the front end of the service and how functions would be grouped within the service's interface. Early prototypes consisted of shallow back-end models and basic ideas about how the front-end would overlay the included functionality. This basic prototype serves as a checklist for functionality required by any established time bank (i.e., any functionality required during the service exchange process) (see **Figure 4.1**). This checklist has been heavily and consistently referenced during design and implementation of NRV Time Bank's system to prevent the neglect of essential time bank functionality in favor of focusing solely on interface design and implementation. NRV Time Bank desires inclusion of all demographics within the NRV community, and a highly usable interface is a means to that end, especially considering the broad age range for potential users. Despite this, having access to an outline of required functions served to keep design and development from deviating too far towards the front end of the system.

Time Bank Site Currently						
Features Overview	Account	Offers	Requests	Members	Groups	Inter-Trade
offer directory	Member Info	see requests*	Search member requests	member list	announce/email group	populate list of other time banks
request directory	Exchanges Made		shows # of requests below it	member since	create group	link list to other banks' sites
statement	Achievements			last exchange data	join group	*hours button disappears
Profiles	Member Bio			location	announcement list	
Endorse	Offers					
Likes (see endorsements)	Requests					
hours / credit	Profile Picture					
	Contact					
	Group					
	My Requests					
	Add Requests					
	My Offers					
	Add Offers					
	Edit Profile					
	Edit Contact					
	Change Password					
	See Statement					
	My Attributes					

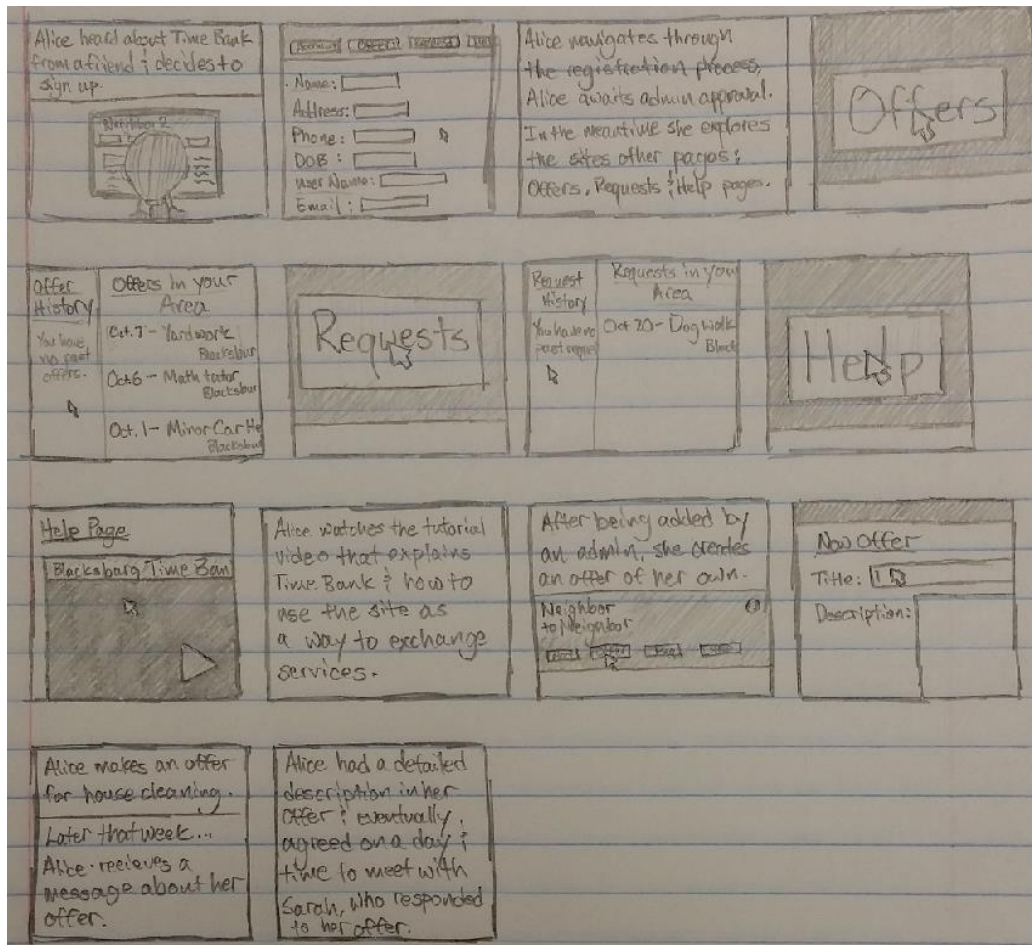
**Figure 4.1:** Early prototype of functional component structures

<https://docs.google.com/a/vt.edu/spreadsheets/d/1wZHB0TbI-4KEaobPALN12OkG5tWN9bfgdZTtVH7TzVI/edit?usp=sharing>

Early prototypes are accompanied by storyboards and scenarios (see **Figures 4.2 & 4.3**) that serve to assist in the exploration of all necessary time bank functionality and its practical application within NRV Time Bank’s system. Despite the functional models of established systems being made available, independent discovery of required functionality was considered necessary so development could proceed with the support of a thorough understanding of why certain functionality is necessary and in what situations. With a deeper insight into why these functions were necessary or desirable, development could proceed without deviating from the primary goal of increasing usability for all stakeholders of the system.



**Figure 4.2:** Early storyboard outlining basic transaction process from the user’s point of view



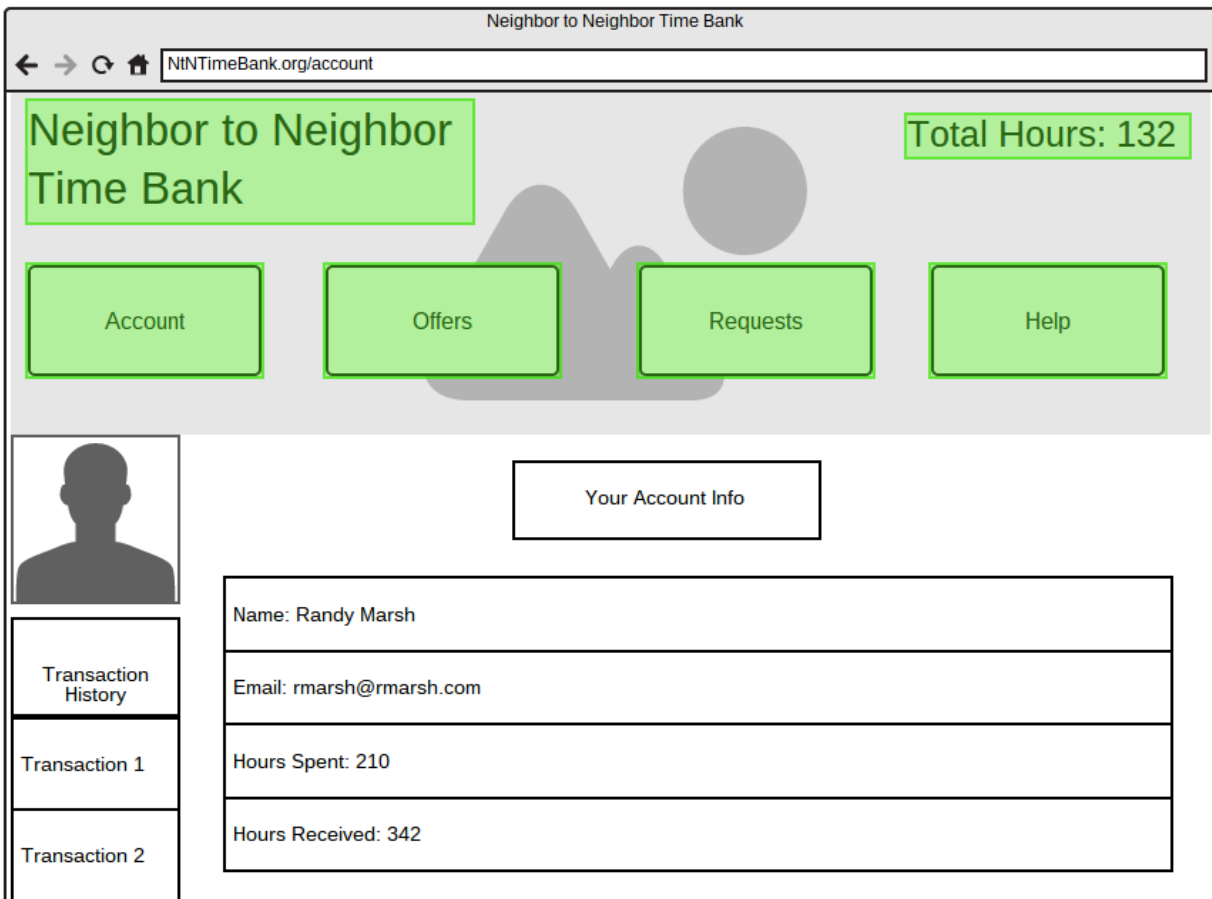
**Figure 4.3:** Storyboard outlining transaction process with more focus on navigation and end result in regards to user

Understanding the reasoning behind necessary TimeBanks functions proved very beneficial during the planning phase of implementation and the implementation process itself. During planning, prototypes used in conjunction with interview notes regarding use of established systems and feelings towards TimeBanks' stated ideals (**See Appendix 10.2.1 & 10.2.2**) assisted in the assignment of meaningful priorities to different groups of necessary functionality. Early prototypes also served to reinforce ideas regarding functionality already formed during storyboarding, interviewing, and analysis of current systems; and helped in the formation of basic outlines for navigation and workflow.

### 4.3 Front End Prototypes

To assist in working towards NRV Time Bank’s stated goal of increasing usability, later prototyping was heavily focused on the front end of our service. Front-end prototypes were made to roughly map out navigation and workflow of required user tasks. Early prototypes were referenced frequently to ensure we covered all necessary functionality. Guidelines for user interaction were created based on functionality requirements established by earlier prototypes (see **Figure 4.1**).

Following is a screenshot of an interface prototype. (see **Figure 4.4**)



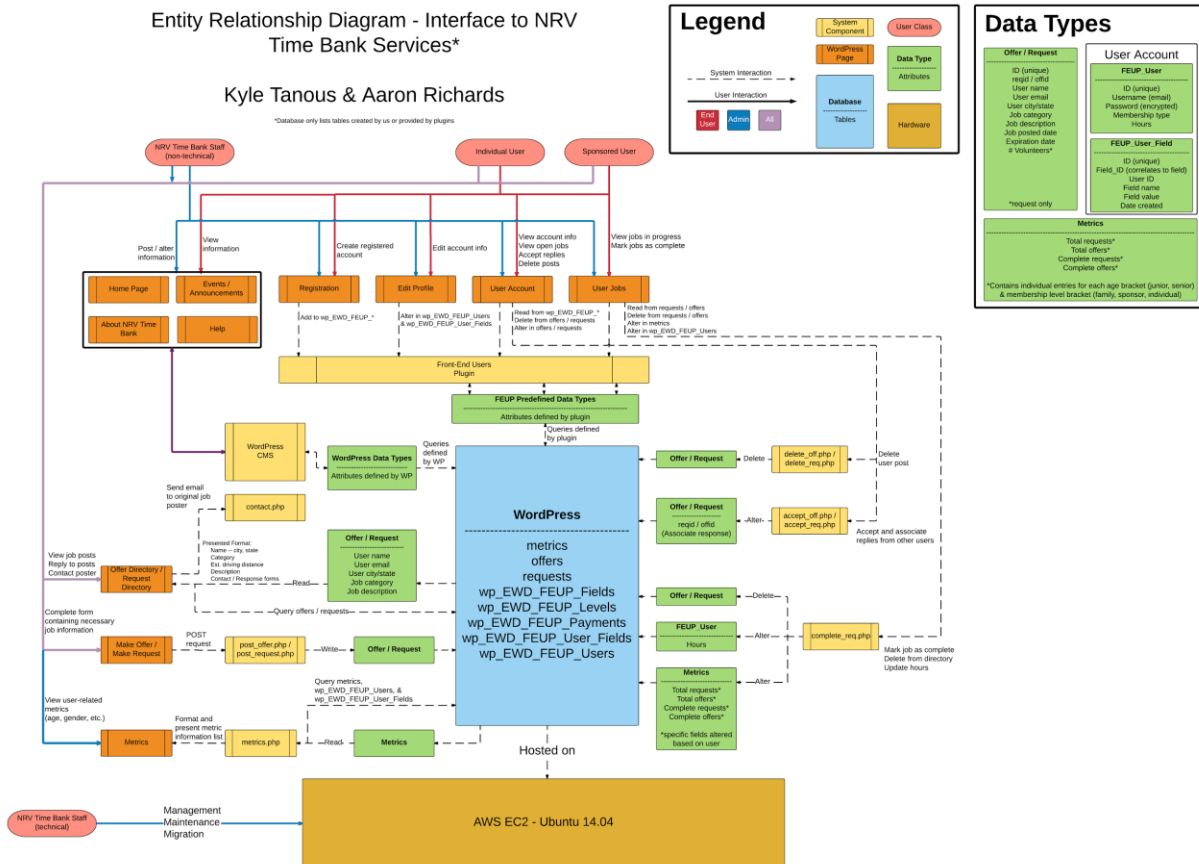
**Figure 4.4:** Moqups prototype with clickable areas highlighted, used in user interaction outline  
<https://moqups.com/ktanous@vt.edu/VHIO1OVe/p:a63d94323>

These interface prototypes proved valuable when developing early iterations of NRV Time Bank’s service, especially with regard to obtaining preemptive user feedback before development began. More focused feedback could be gathered by presenting an interactive prototype to users and clients. An interactive prototype allowed more targeted questions about interaction experience to produce usable feedback during interviews.

Front-end prototyping decisions were based primarily on user feedback, which can be noted by the emphasis on easy navigation and prominent information display. User feedback was primarily gathered from a younger, college-age crowd, despite NRV Time Bank's goals of inclusion among the elderly citizens of the community. The reason for this is one of necessity over ideals. Feedback gathered from senior residents of the NRV community lacked direct application during development due to a widespread lack of general computing experience or knowledge among subjects. Despite the lack of direct application, feedback from senior citizens expressing interest in alternatives to the web-based approach being developed reinforced the necessity of high usability and learnability with NRV Time Bank's web service if it was to serve a role in helping to reach stated goals.

## 4.4 Prototype to Early Implementation

Transitioning from prototyping to implementation solidified roles of prototypes as reference points. Referencing of prototypes during implementation went in similar order to prototype development. Early functional prototypes were referenced heavily during back-end PHP script development. It was also during this time we set up the standard WordPress MySQL database with new tables to accommodate our needs. (see **Figure 4.5**)



**Figure 4.5:** Entity relationship diagram for NRV Time Bank system. Shows the interactions between Time Bank staff, users, and the various pages and data types our service contains. The diagram can also be at: <https://www.lucidchart.com/invitations/accept/28dfe6dc-e568-4f43-bb84-608281c42ee3>

Front-end prototypes were referenced in conjunction with functional prototypes during interface development. Notable aspects of the site's homepage (see **Figure 2.1**) derived from prototypes include the persistent top bar menu suggested by our Moqups prototype, as well as an announcement feed. NRV Time Bank staff feel having these announcements prominently displayed on the home page will help foster the sense of community and coherence they want to achieve with this project by providing a centralized hub for learning about community events.

## 4.5 Conclusion and Moving Forward

---

In conclusion, prototyping proved to be a highly educational process. Despite previous prototyping experience of the development team, the timeline associated with the development of NRV Time Bank's software system forced rapid development of prototypes that retained focus on key aspects of functionality and user interaction experiences. Allowing prototype development to roughly coincide with implementation timelines helped to ensure a smooth transition to implementation. Separation of back and front-end prototypes allowed further exploration into options available for functional and interactive system components.

Moving forward, plans include development of more detailed prototypes for specific functionality, such as real time data feeds for users; and convenience-focused functionality, such as live online chat and support. Future prototypes should continue to focus on improving user interaction experiences, with the ideal end product being a self-contained hub for the exchange of services and information regarding community events. The hope of the development team is that this template, along with the resources we have used and produced, will serve to assist other established time banks create more robust and flexible interfaces to their services.

## **Section 5 Testing**

### **5.1 Introduction**

---

Testing of NRV Time Bank’s software system is largely manual and qualitative. Early testing evaluated the correctness of implemented functional system components and associated user feelings and experiences to gauge the usability of the interface. This method of testing was employed throughout development of the system to retain the presence of effective reference points. Testing of later iterations maintained similar goals of early testing procedures with the additional goal of capturing quantitative metrics such as number of navigational steps and clicks performed during specified tasks.

### **5.2 Functionality Testing**

---

Manual testing was used to evaluate early iterations of the system. Carrying out the transaction process ourselves and manually checking the database to verify correctness greatly increased our own understanding of the process and what improvements might be made. A notable benefit of this evaluation method, when combined with qualitative evaluation of the system’s interface, is the ability to consistently create optimized short-term goals and guidelines for implementation. Generally, manual testing was carried out using several developer accounts that could stage service exchanges with impunity.

### **5.3 Usability Testing**

---

#### **5.3.1 Qualitative Evaluation**

Qualitative evaluation of the interface to our service takes the form of feedback from interviewees of widely ranging ages, occupations, and levels of computer literacy. The interviews were conducted with usability in mind. Potential interview questions required users to perform specified tasks, such as registering for an account. Evaluation covered the user’s feelings and the correctness of the task performed.

This feedback helped in forming future design decisions. In the earliest stages of development four interviewees were asked for feedback regarding the hOurworld interface. The feedback was negative overall, and used to define problem areas in navigation and task procedures. After the interface to NRV Time Bank’s system could be considered functional, we asked four more interviewees to compare the two interfaces and perform basic tasks. These tasks were monitored to record any apparent struggles encountered during the process.

Once our site was equipped with the necessary functionality, we engaged another set of four interviewees to gauge their ability to use NRV Time Bank’s site. Most subject from this

interview were able to carry out common tasks requested of them, including account registration and posting a job without assistance.

### 5.3.2 Quantitative Evaluation

Quantitative evaluation of our interface required the drafting of standardized metrics able to be recorded. We eventually settled on the following three metrics we deemed feasible to record. (see Table 5.1)

**Table 5.1: Quantitative Evaluation Metrics**

- Number of navigational steps performed during a specific task
  - Compares number of actions performed by user to the optimal number of steps (i.e., the minimum number of possible navigational clicks that may be performed during a specified task)
  - Allows comparison of differing interface navigation components to determine most optimal presentation of navigation options
- Time required to finish a specified task
  - Recorded without accounting for variables such as typing and reading speed
  - Serves as rough measure of learnability when comparing required time across multiple task performances
- Correctness of specified task performance
  - After a task is performed, correctness of values input by the user are manually evaluated
    - Registration fields - ensuring fields are correct (e.g. ensure user's "city" field contains only a city value, and nothing else, such as state)
    - Description of job posting - determine whether or not description is suitable
    - Category of job postings - ensure category is an appropriate match for the description
    - Email submission form - ensure subject line and description are appropriate for the specified task
    - Job completion form - ensure value of hours worked is appropriate for specified task
  - Helped make poor documentation and explanations apparent (e.g., updating the Help page to clarify service exchange procedures); results used in planning of future interface changes to increase learnability

## Section 6 Manuals

### 6.1 User's Manual

---

#### 6.1.1 Introduction

NRV Time Bank's website can be found at <http://52.32.67.219>. The majority of site navigation may be performed by clicking on the links provided in the top bar menu, which is present on all pages. To reveal more options available from the top bar menu, hover your mouse over menu items. On the menu at the top of the webpage, there is a section called "Help". This manual will cover the same material as that help section, but more thoroughly.

#### 6.1.2 Account Registration and Login

The homepage of the website consists of navigation components that lead to all areas of the site. Member login fields can be found at the top of the home page to the right of NRV Time Bank's logo. If you do not have an account, you can register for one using the "Register for Membership" item located within the top bar menu. Upon registration you will be asked to enter information into the following fields.

- Email
- A password to be used at login
- Name
- Gender
- Membership type
- Date of birth
- Phone number
- Address and zip code

After completing the form your account can be registered. Information you entered during registration can be viewed by accessing the "My Account" page accessible through the top bar menu.

#### 6.1.3 Posting Content

On the top bar menu, there are two entries labeled as "Requests" and "Offers". By hovering your mouse over either one of these items, an option will be revealed titled "Make Request/Offer" that will lead you to a submission form for posting a request or offer. The submission forms for offers and requests are identical, except for requests offering an option for number of requested volunteers. The submission forms for offers and requests let you select category and expiration date, as well as provide a description of the offer or request you are posting.

#### **6.1.4 Viewing and Interacting with Content**

Directories of posts created by users can be accessed similarly to accessing submission forms for content. Hovering your mouse over the “Requests” or “Offers” items on the top bar menu reveals links to these directories. These directories may also be reached by simply clicking the “Requests” or “Offers” menu items. Each entry in the directory contains drop-down forms for responding with a post of your own, or sending an email directly to the original poster. These forms can be opened by clicking on the “Send Email” and “Make Offer/Request” located at the bottom of each individual entry in the directory.

You can view your own posts by clicking on the “My Account” item located in the top bar menu. This page contains account information, and any open jobs you have posted. A job is considered open if a response has not been accepted. Clicking on the “Offers/Requests” drop down menus, found in every job entry, reveals any posts other users have made in direct response to yours. Any of these responses can be accepted by clicking the “Accept Offer/Request” button associated with the response you wish to accept.

Upon accepting a response, a job is considered in progress, and can be viewed on the active jobs page. This page can be accessed by first hovering your mouse over the “My Account” item in the top bar menu and selecting “My Jobs”.

#### **6.1.5 Completing Jobs**

Active jobs can be marked as complete by the original requester of the job. Upon completion, the requester enters the number of hours worked. The number of hours and jobs completed for the involved users is updated, and the job ceases to exist.

## **6.2 Developer’s Manual**

---

### **6.2.1 Back-End Structure**

The back-end of our service is handled primarily by WordPress and its plugins. Options and documentation for these can be found at the following links:

- WordPress - <https://codex.wordpress.org/>
- Front-End Only Users - <https://wordpress.org/plugins/front-end-only-users/developers/>
- PHP Code Widget - <https://wordpress.org/plugins/php-code-widget/developers/>

These plugins handle user accounts and provide additional flexibility beyond what is normally available through WordPress.

Back-end features we have implemented ourselves primarily consists of PHP scripts that read and write to databases. The MySQL database containing all site content is hosted on the server, and is part of the standard WordPress installation. A list of these scripts and the provided

functionality follows (if a script is not listed as a WordPress page, then it can be found in /var/www/wordpress/ on the site's server). A relationship diagram showing how these scripts interact with the WordPress database and its pages can be seen in figure 4.5.

- `accept_off.php/accept_req.php`
  - This script runs when a response is accepted by the original poster of a request or offer. When accepted, the response's database entry is altered to include a pointer to the original post so the two can be associated with each other.
- `post_offer.php/post_request.php`
  - This script runs when a user submits an offer or request form through their respective pages, or when a user responds directly to another user's post. In the case of form submission from the job creation page, the offer or request contains no pointers to other posts. As expected, a post made in direct response to another is created with a pointer to the original post.
- `delete_off.php/delete_req.php`
  - This script runs when a user decides to delete one of their posts. Simply removes the offer or request's database entry.
- `complete_req.php`
  - This script runs when the original requester of a job marks the job as complete. Hour and jobs completed counts are updated for involved users and both the offer and request are removed from the database.
- `contact.php`
  - This script runs when a user sends a message to another user. The script sends an email to the original poster in the case additional contact is required before a job can be started. This feature is unavailable until a domain name for the hosting server can be registered. This feature will work when the site has a proper domain.
- `metrics.php`
  - This script is for administrative purposes. It can be accessed at `52.32.67.219/metrics.php`. The script presents user-related metrics (e.g. new senior/junior members this month) and presents them in an easily read format for reporting to the TimeBanks organization.

## 6.2.2 Front-End Structure

Source code for the front-end of our service is available through the WordPress CMS. Access to the WordPress CMS can be obtained by logging in with an administrator account at `52.32.67.219/wp-login`. The list of pages and the functions they provide are as follows:

- Edit Profile
  - Provides a form that allows users to edit the profile information that was submitted upon registration.

- Help
  - Provides text instructions to assist in the performance of tasks involved in the service exchange process.
- Make Offer/Request
  - Provides submission forms for the creation of a new post. Passes parameters entered by users to `post_offer.php/post_request.php` scripts to be added to database.
- My Account
  - Presents users with their account information and their open jobs (posts which have not accepted a response). Here users are able to accept responses and delete their posts.
- My Jobs
  - Presents users with their active jobs (posts for which a response has been accepted). If the user has active requests, they are given the option to mark as complete, which will update hour and job counts for the involved users via the `complete_req.php` script.
- Offers/Requests
  - Directory pages for offers and requests. Here users can browse posts, sorted by either category or date posted. Any changes to sort option when viewing the list of Offers/Requests performs a window replace instead of a redirect to keep back button navigation intuitive (e.g., if a user selects a new sorting filter, use of the back button will not navigate to the same page with the previous sorting filter, but instead to the most recently accessed page before navigating to the directory page).

### 6.2.3 Migration and Maintenance

Migration of the service can be achieved through AWS imagine services. In the case that the service is no longer being hosted on an EC2 server instance, the WordPress installation can be transferred through standard means as outlined in their guide ([https://codex.wordpress.org/Moving\\_WordPress](https://codex.wordpress.org/Moving_WordPress)), or by means of a plugin (<https://wordpress.org/plugins/all-in-one-wp-migration/>). Regardless of the transfer scenario or environment, some care must be taken to ensure all links retain proper destinations within the service.

Because the WordPress service serves as the foundation for most of our service's functionality, most troubleshooting can be performed by following the WordPress guide (<https://codex.wordpress.org/Troubleshooting>). Any issues relating to our service in particular will likely be related to database configuration. See Figure 4.5 for database configuration and custom data types available

## **Section 7 Reflection**

### **7.1 Timeline**

---

Early on in the project's lifecycle, the group's planned timeline was fairly volatile, due to external requirements put in place by the TimeBanks organization. These requirements usually required a somewhat functional implementation of our service to be displayed to outside stakeholders of our service. This certainly proved to be a new challenge to the group. Premature implementation based on a less than ideal amount of feedback and analysis had to be corrected several times to get back on the proper, planned development cycle.

Outside of events beyond our control, our development process managed to roughly follow timelines set forth by our client early in our involvement with the project. Information gathering and analysis were followed by design, and finally development. After development began, iterative improvements were made to our service based on referencing of new and old feedback and analysis as required.

### **7.2 Problems**

---

We experienced several new challenges during the course of this project. Most prominent was the lack of directly applicable feedback provided by senior citizens, who represent the intended beneficiaries of NRV Time Bank's web service. This lack of information served to complicate a development cycle we had become accustomed to. We were forced to extrapolate based on information lacking relevancy and reference text materials more than we have in the past.

### **7.3 Solutions**

---

The lack of directly applicable feedback from the senior residents of the NRV was solved by a shift in development focus. Instead of designing for a single demographic, we changed our approach to focus on increasing usability across all demographics. This solution was born out of necessity, but taught the group an important lesson about the importance of improvisation and the place it holds in an effective development process.

## Section 8 Future Work

Future work will consist mostly of the same iterative development cycle (see **Table 8.1**), but implementing new functionality. Future work exists outside of our group's scope of expertise. Therefore, outside of implementing new functionality, future goals also consist of bringing in new team members to cover our weaknesses. The most notable weakness of our group is a lack of an exceptionally thorough understanding of WordPress and its potential. While we have certainly managed to create a website that is at least somewhat aesthetically pleasing, we believe working with somebody who could leverage the power of WordPress to its fullest extent would certainly speed up and ultimately improve implementations of our service.

A secondary concern for the future consist of improving hosting hardware/migrating to a local hosting environment. This work is contingent on available funding to obtain and migrate to new hardware, and staffing so that a locally hosted service can be effectively maintained.

### **Table 8.1: Future Goals**

- Register a domain
  - Allows user-to-user and admin-to-user contact via email
- Interactive information delivery
  - Real-time feeds
  - Chat
- Rich media in help and about pages
- More thorough documentation for users and administration
- Complete another set of interviews with the current iteration of the service
  - Groups can be assembled by NRV Time Bank staff
- Complete handoff of the service to NRV Time Bank administrators
  - Migrate to local hosting hardware

## **Section 9**

### **Acknowledgements**

Zoe Belyavsky (zbelyavsky@blacksburg.gov), our client, along with Ellen Stewart have been ever-present sources of guidance during development. Her insight into requirements set forth by the TimeBanks organization proved invaluable throughout the development cycle in ensuring we delivered a suitable software system to the NRV Time Bank.

## **Section 10**

### **Essential Interview Notes**

Following is a compilation of essential formal and informal interview notes that served as key reference points during design and development. Interviews were all conducted by Aaron Richards and Kyle Tanous. Interview sessions attempted to include residents of various educational and age demographics when possible to maintain relevance to NRV Time Bank's desired outcomes of community inclusion. Notes have been abstracted into key points so as to provide complete anonymity to interview subjects.

#### **10.1 Senior Residents of NRV**

---

Early interviews with senior (60+) citizens of NRV, before functional iteration of web service was available. Subjects were only presented with TimeBanks' philosophical ideals and abstract outline of practical service usage.

- Widespread lack of general computing knowledge
- Lack of desire to learn web-based timebank service
  - Willing to use phone or mail based service
- General disdain for philosophies and ideas behind TimeBanks
- Widespread lack of PC ownership and basic (e.g., email) knowledge
- Reliance on grandchildren for help with general computing
- Minority familiar with old programming languages (e.g., Fortran) and appreciation of quality software systems

#### **10.2 hOurworld User Experiences**

---

Intermediate interviews with subjects possessing experience ranging from undergraduate student to working professional. Subjects were presented with Moqups prototype and TimeBanks' philosophical ideals.

- Majority of subjects experienced trouble performing basic tasks such as posting and responding to job requests
  - Assumption is that trouble was caused by lack of readily accessible assistive text
- General confusion in regards to capabilities of software system
  - Assumption is that confusion is due to intended task procedures not being clearly explained
- Minority of subjects able to easily perform basic tasks required during the service exchange process

- Likely due to relatively high levels of computing experience possessed by this minority
- Within minority, widespread notice of lack of readily available help section

### **10.3 Early Implementation Impressions**

---

Information gathering for early fully-functional iteration of our service from people of relatively technical backgrounds, mostly graduate students. Subjects were presented with iteration of service possessing fully capable back end, but slightly unrefined interface, which lacked emphasis on assistive text and clarification of intended task purposes.

- Widespread disdain for bland aesthetics
- Subjects involved in UI design offered suggestions for referencing proper design guidelines
- Widespread sentiment that navigation could be further simplified
  - Mostly in regards to page switches (e.g., some subjects commented they should be able to view requests and offers on same page without navigating to new page)
- Remembrance of actions trumped recognizability
  - Recognizability of available actions negative, first task completion time noticeably worse compared to task completion times of repeated task procedures
- Subjects experiencing confusion seemed to refuse to access help page

### **10.4 Intermediate Implementation Impressions**

---

Information gathering for fully-functional iteration of our service boasting more developed interface than previous interview session. Subjects represented age demographics from early-twenties to mid-sixties. Subjects were presented with iteration of service altered based on feedback from early implementation impressions.

- Fewer issues performing content posting and interaction tasks compared to earlier interview sessions
- Most prominent issue involved recording completion of service exchange
- Minor complaints regarding aesthetics, particularly of directory pages
- Relatively positive overall attitude regarding interface
  - Switch in WordPress theme from early iteration

## Section 11

### References

Hartson, H. Rex, and Pardha S. Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Amsterdam: Elsevier, 2012. Print.

"HOurworld - Time Bank Mobile Web Services." *HOurworld - Time Bank Mobile Web Services*. N.p., n.d. <http://hourworld.org/> . 23 Mar. 2016.

"HOurworld Directory of TimeBanks, Community Directory: An International Network of Hour Exchanges (time Banks)"" *HOurworld Directory of TimeBanks, Community Directory: An International Network of Hour Exchanges (time Banks)*" N.p., 2006. <http://hourworld.org/>. 02 May 2016.

"Community Weaver | TimeBanks USA" *TimeBanks USA*. TimeBanks USA, 2014. <http://timebanks.org/get-started/community-weaver/> . 02 May 2016.

"Go Deeper | TimeBanks USA" TimeBanks USA. TimeBanks USA, 2014. <http://timebanks.org/go-deeper/> . 03 May 2016.