Behavior App

Gamifying Good Behavior

Developers: Lawrence Glick, Richard Patten

Clients/Co-workers: Loran Hauserman, Josh Hauserman

Instructor: Edward Fox, CS 4624, Multimedia, Hypertext, and Information Access

Virginia Tech, Blacksburg VA 24061

05/7/19

This report is not confidential

Table of Contents

1. Introduction

There is a misconception that gaming and societal excellence are opposing forces; A child or young adult that spends time gaming is spending time away from study and chores. We see this disconnect in the interactions of parents, school districts and students as motivation for creating Learn2Game, Odjin LLC's prototype behavioral therapy focused mobile application.

Students can over-prioritize games while parents may neglect to understand and nurture their children's hobbies, which can ultimately result in declining academic or personal performance. Instead, we see a centralized point of contact by which to establish video games as a motivating tool rather than an escape from responsibility. Children and young adults - Gamers - and their parents and authority figures - Coaches - will be able to use our app to help alleviate the conflict around balancing perceived good and bad behavior relating to video games. Over time, this will work to create dialog where the Gamers see Quests as fun opportunities for rewards in the form of video games, and Coaches see Quests as opportunities to instill good behavior. Odjin's app will allow parents to stay informed on their children's gaming habits by showing them simple ratings and descriptions of the games their children are playing and want to play. Parents can choose to reward their children with games they feel safe with and then know their children engaged with content appropriate and entertaining for them.

2. Executive Summary

For Spring 2019 for CS 4624, Multimedia, Hypertext, and Information Access, our team consisting of Lawrence Glick and Richard Patten was tasked as members of Odjin LLC, our client, to begin development of Learn2Game, the company's startup android application.

Over the course of the semester, we developed an android app fulfilling the function of the requirements of the project. This app was built from the ground-up as a new project in Android studio in Java, utilizing four platforms working together: Android Studio, Firebase Authentication, Firebase Live Database, and a game distribution API designated in the developer's manual.

Due to the complexity of android app development and its approach in using runtime/multithreaded event handling, we initially estimated development would take most of the semester, with enough time to conduct user testing and release at the end. We found that completing every feature the clients wanted would be an overestimation, as we could not balance our other responsibilities as a two person team and still complete every feature that was promised at the beginning of the semester. By the end of the semester, we were able to complete the Authentication system, Main Screen, Quests, WishList, Profile/Networking features of the app. Transactions, user testing, UI/UX review, and later product related work highlighted in section 9, future directions, will be completed throughout the rest of 2019, since the development team are also stakeholders of the project's success.

This is an independent project owned by Odjin LLC, and no Virginia Tech development resources were used in the creation of this project or its materials.

Mission Statement

> Utilising ABA principles, we will incentivise academic and personal success through our task-reward based app.

3. Project Requirements

Project will be accessed by the client on the completion of the following criteria:

3.1 - Authentication System - Develop an authentication system that allows for users to create accounts, log in, and track their profile information

Goals:

a. Unique Account Registration - User can create a unique account through input of a username, email, and password
b. Database information - user's quest/wishlist data is stored remotely and retrieved upon logging in.
c. Live updating - A core philosophy of our app is that we want live updating of a user's information as they recieve it (ex. Once a quest is submitted the recipient can immediately see it appear in their quest list)
d. Log in - User can log into their created account
e. Log Out - User can log out and return to the Login screen

3.2 - Connect Accounts - Develop a networking system that allows a user to link other accounts to send quests.

Goals:

f. Profile page - A user can navigate to their profile page where they can log out and process the following features.
g. Send invite - A user can send a link request to another registered user
h. Receive invite - Once a request is sent, the invite appears at the top of the recipients' linked accounts list as a pending invite
i. Accept or deny invite - Buttons for accepting or denying the above request appear accordingly, which the recipient can either accept or reject the invitation to link accounts.
j. Linked accounts - Once the recipient has accepted the invitation, both users can assign quests to each other and view each other in their profile page.
k. Remove link - Give the user the ability to remove the link between accounts

3.3 - Main Screen - The purpose of the main screen is to display all necessary information related to the function of the app upfront to the user once they are authenticated.

Goals:

l. Quest card list - A list of active quests are displayed at the top of the main screen of the app.
m. Game wishlist - A list of games that have been added to the user's wishlist is displayed second

n. Most popular - A list of game suggestions for the 10 most popular games are displayed at the bottom of the main page
o. Access - The following fragments can be accessed from the main page: Profile, Quests, Wishlist.

3.4 - Quests - This is the behavioral psychology page of the app. Here, a user can access the below functions for assigning and completing quests.

a. Quest list - A list of a user’s assigned active/inactive quests is displayed on this page.
b. Create a Quest - User can build a quest only for connected accounts, and once it is submitted it appears in the recipient’s list of quests. This is reflected on the recipient’s main screen.
c. Complete a Quest - The user can mark a quest completed, where the attached user is notified of the completion of the quest.

3.5 - Wish-list - This page of the app allows the user to add potential rewards for completing their quests.

a. Search for game - The user can search through the list of games by accessing the API.
b. Add game - The user can add a game to their wish-list, which is reflected on the main screen.
c. Remove game - The user can remove a game from their wish-list
d. If time allows, user can view connected accounts wish lists.

4. Design

As part of our app’s design philosophy, we wanted to maintain a consistent UI while changing out the information displayed based on what the user wants to view. To accomplish this we used an activity-fragment layout. This allows us to maintain an activity that responds to callback events and changes fragments, and fragments to display information, initiate callbacks, and accept user input.

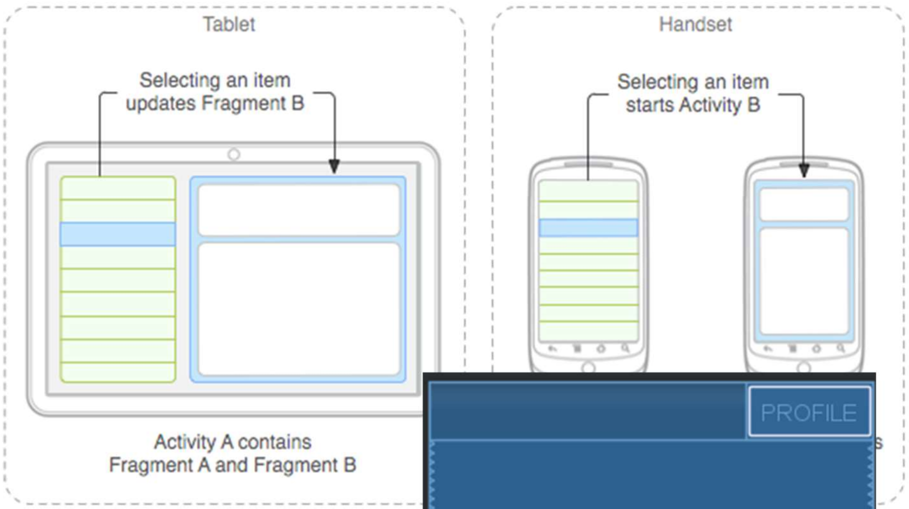

Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B

Fig. 4.1 - An example of how fragments are exchanged within activities

Fragments also allows us to maintain flexibility in our UI, as we can adapt which are displayed depending on the orientation or type of the device it is being used on. Our app's main layout involves a netflix-style layout, shown in the blueprint to the right, to have consistent buttons available at all times to the user. The center panel is interchangeable, and will switch out with the following 4 families of panels, which populates on startup with information needed.

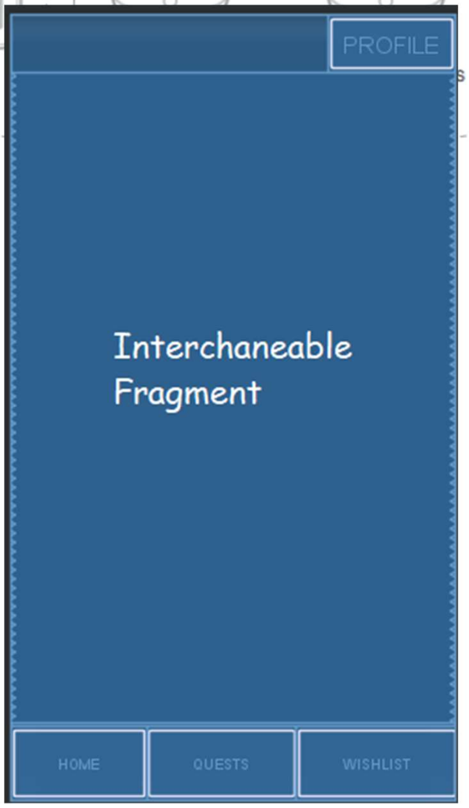
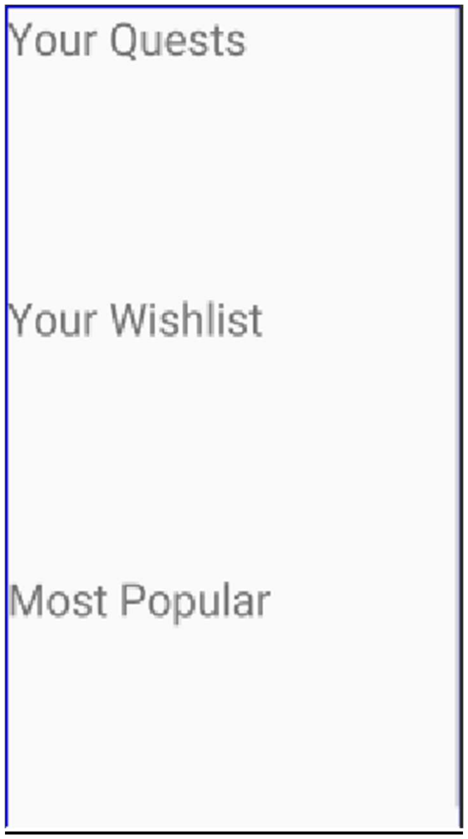
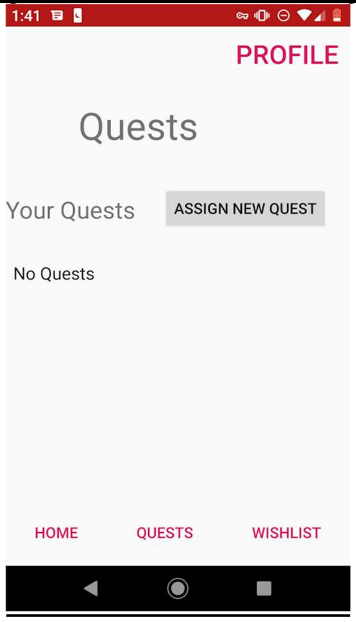
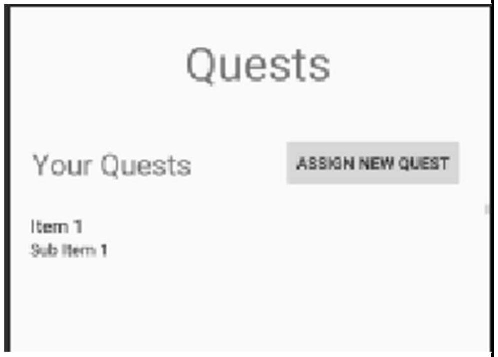
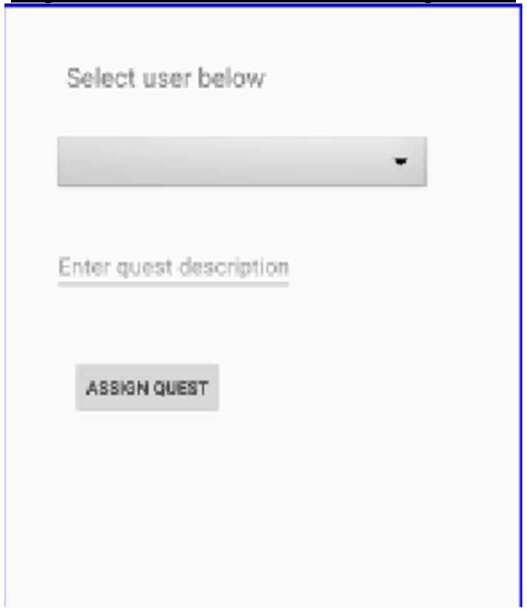
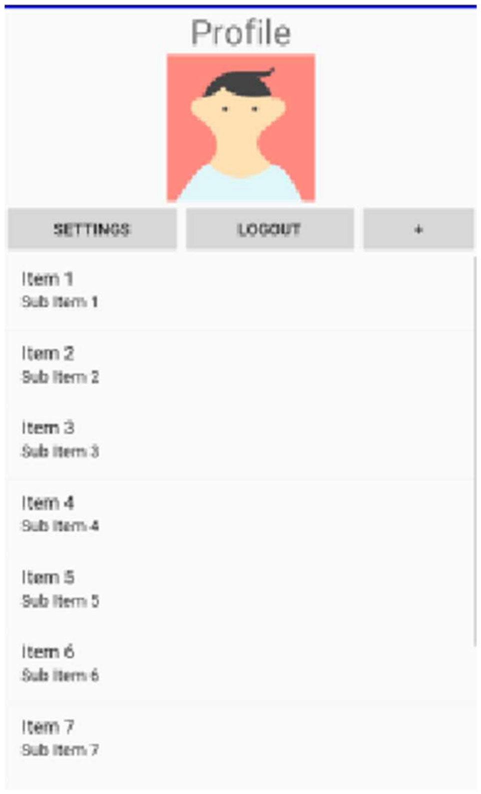


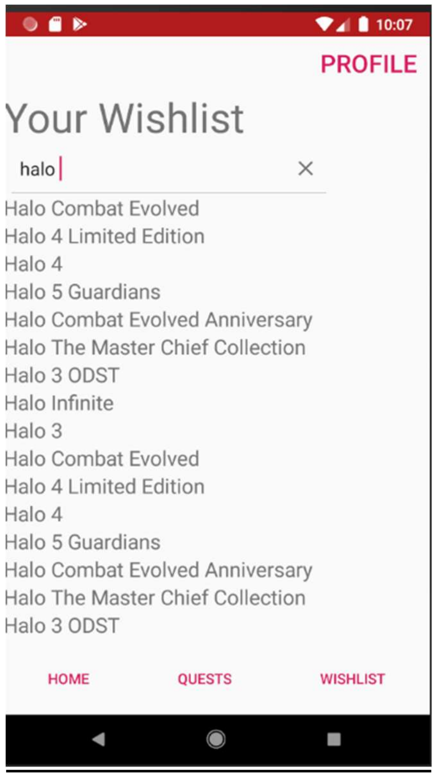
Fig 4.2 - LoggedIn Activity

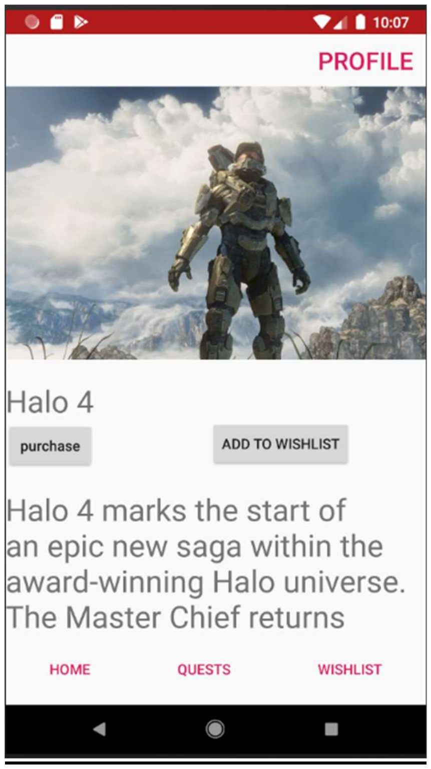
Panels & Screenshots

1. Home Fragment The Home Fragment is the main fragment that is presented to the	Fig 4.3 - Home Fragment

user on the app's startup and when they seek to return to the main page of the app. This includes a short overview of the three sections we consider to be the most important to our app: quests, wishlist, and a short storefront.	
2. Quest Fragment This is a collection of 2 fragments that can be exchanged depending on whether the user is viewing quests or building a quest to assign. a. Main Quest Page This fragment is used to display necessary quest related information to the user.	Fig.4.4 - Main Quest Page  1:41 PROFILE Quests Your Quests ASSIGN NEW QUEST No Quests HOME QUESTS WISHLIST

b. Quest Builder When prompted by the  Quests Your Quests ASSIGN NEW QUEST Item 1 Sub Item 1 user, the app switches to this fragment to allow the user to assign a quest to another valid user registered in their friends list.	Fig 4.5 - Quest Builder Fragment  Select user below Enter quest description ASSIGN QUEST
3. Profile Fragment The profile fragment allows for linking accounts, receiving requests, logging out, and other profile-related functions. Space is reserved for a profile picture once it is implemented.	Fig 4.6 - Profile Fragment  Profile SETTINGS LOGOUT + Item 1 Sub Item 1 Item 2 Sub Item 2 Item 3 Sub Item 3 Item 4 Sub Item 4 Item 5 Sub Item 5 Item 6 Sub Item 6 Item 7 Sub Item 7

4. Wishlist Fragment The purpose of this fragment is to display to the user all the information necessary to add a game to their wishlist. It includes a search bar to look for a game to add, and below a dynamically resized list of games the user currently has in their wishlist.	Fig 4.7 - Wishlist Fragment  PROFILE Your Wishlist halo Halo Combat Evolved Halo 4 Limited Edition Halo 4 Halo 5 Guardians Halo Combat Evolved Anniversary Halo The Master Chief Collection Halo 3 ODST Halo Infinite Halo 3 Halo Combat Evolved Halo 4 Limited Edition Halo 4 Halo 5 Guardians Halo Combat Evolved Anniversary Halo The Master Chief Collection Halo 3 ODST
5. Details Fragment This fragment is accessible by any other fragment that views games. This displays game information such as the title, description, a scrollable window of screenshots, and ESRB. It includes a button to add to wishlist, and a (dummy) button to purchase.	Fig 4.8 - Game Details Fragment

	 10:07 PROFILE purchase ADD TO WISHLIST HOME QUESTS WISHLIST Halo 4 Halo 4 marks the start of an epic new saga within the award-winning Halo universe. The Master Chief returns

5. Implementation

5.1 - Public Information Implementation

Project management was handled by Loran Hauserman, who coordinated design efforts of our subject matter expert, Josh Hauserman, and the development efforts of Richard Patten and Lawrence Glick.

Trello was used to monitor and keep track of the completion of features and backlog items throughout the development process.

All code was written in android studio using git for version control. Android studio allows us to effectively set up a system to develop android apps, as well as enable testing through android emulation of different OS versions and our own personal devices as well.

For data storage and authentication, Learn2Game uses Firebase as its platform. Firebase conveniently integrates into android apps smoothly, effectively, and securely, as well as allows us to have the flexibility to store whatever data we need. We also chose firebase as it is designed by Google alongside Android, which automatically provides most of the necessary security elements we are looking for. This also enables us to utilize google play transactions to provide customer security and confidentiality.

We do not wish to publicly disclose our API until the full release to protect the company’s goal. The API we used enables two functions of the app:

1. It allows us to pull video game information such as cover art, ESRB rating, description, etc.
2. Allows us to purchase game key and send key to customer.

6. Testing

6.1 Unit testing

Android already contains a testing environment that is set up to access whether the application is ready to submit to the google play store, and meets all of the testing standards set by Google. We haven't looked into detail as to what these tests include, but plan on beginning testing once transactions are fully completed and the UI/UX has been reviewed and finalized.

Google has a set series of goals each test of the app is required to meet to maintain quality of its apps. These include tests for stability, security, battery life, etc. and that it adheres to google play policies. Documentation for what goals the test suite looks for is listed here: https://developer.android.com/docs/quality-guidelines/core-app-quality

6.2 User testing

We have identified a small set of users that would be interested in testing this app, which mostly include family and friends that have children who are interested in video games. Guidelines for user testing such as what questions to ask would be defined by the client, Josh Hauserman, as he is very familiar with wording questions to generate appropriate and focused responses from the user. We are seeking a wider audience for user testing that would give unbiased constructive feedback on what they want to get out of the app, and other possible features that could be added to the project.

So far, we have received consistent positive feedback on the majority of people we have pitched the project both at VTURCS and in passing conversation. The most enthusiastic feedback comes from parents we have met, who see this as another useful tool to help manage an aspect of their busy lives, but also business professionals that see the potential of applying behavioral psychology to the workplace. If the tool exists, it appears that there is an audience that will use it once it's ready and cleaned up.

7. Client Review

On 4/2/19, we met with the client for final evaluation and review of the work we have completed listed in this final report. The app was evaluated based on the quality and completion of the points described in section 3 of this report.

Overall, the client was satisfied with the amount of work completed. They described in our final meeting that their primary concern was the viability of the project and creating a proof of concept. The client's strategic goals when evaluating our app included reviewing viability of:

1. An android app can actualize the vision of the CEO
2. Firebase database and authentication
3. API usage
4. Friend networking

The app demonstrates that all of these functions and are possible to integrate into the final product with our current capabilities.

8. User’s Manual

8.1 User’s Manual

1. Installation

   To install the app once it is released, search for Learn2Game by Odjin LLC on the google play store.

2. Using Learn2Game

   a. If you want to coach others,

   To use Learn2Game as a coaching tool, start off by creating an account by starting the app and clicking on the “register” button. Once you have registered, make sure those you wish to coach have registered as well.

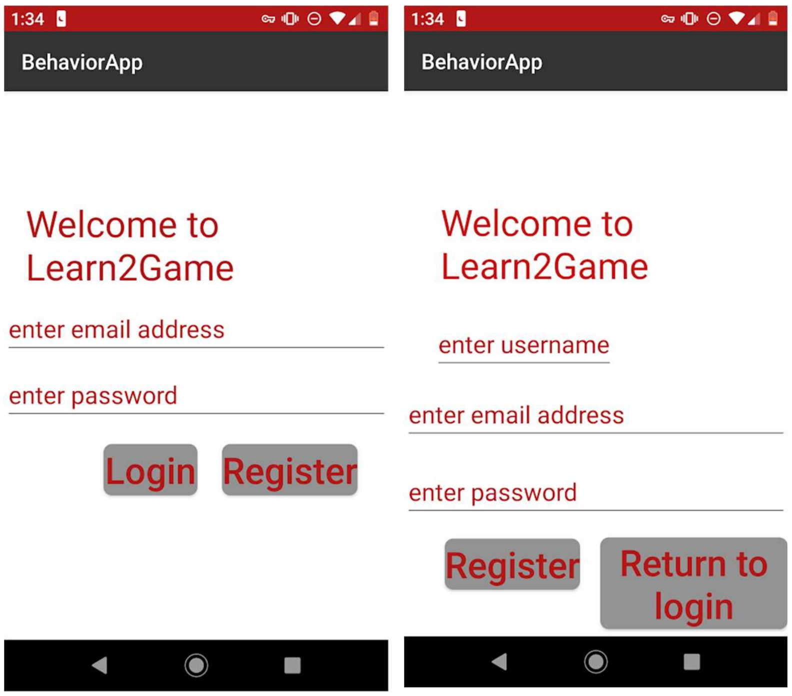
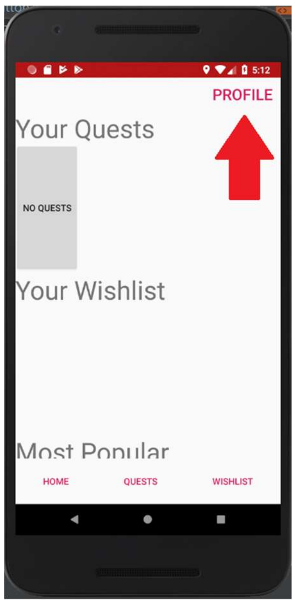


Fig. 8.1.1

Next, enter your email and password to login. Once logged in, tap on the profile button in the upper right hand corner. This brings you to your profile page, where you are able to connect with other accounts.



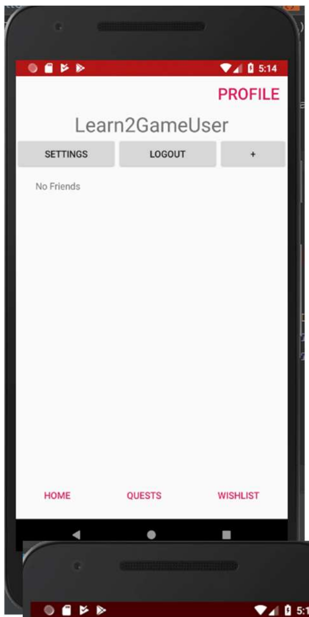


Fig. 8.1.2

Tap on the "+" button above the connected account list, and input the email of a user you wish to connect with. Press enter, and tap the name to send them a request.

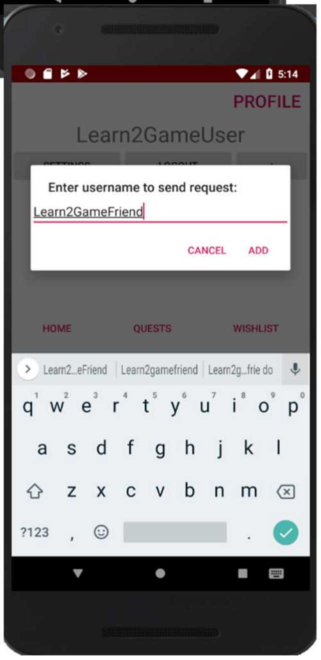
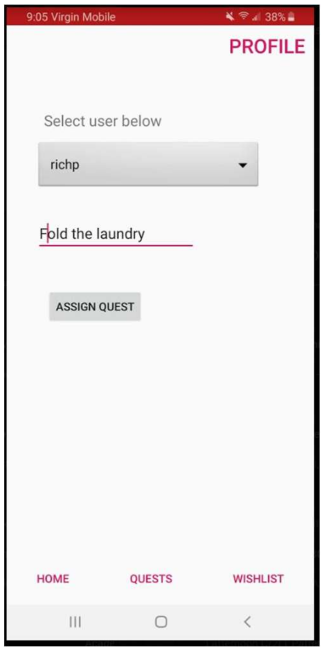


Fig. 8.1.3

Once the other user has accepted the request, go ahead and navigate to the "quests" page on the bottom bar. Click on the "assign" button to assign a new quest. Select a user from the dropdown menu, and specify what you wish to be completed and which game from the user's wishlist you wish to promise. Tap "send quest" to start!



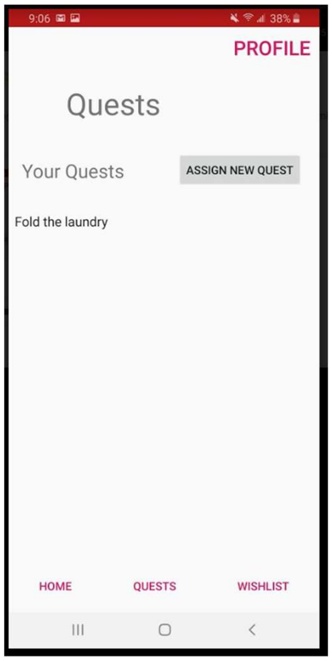


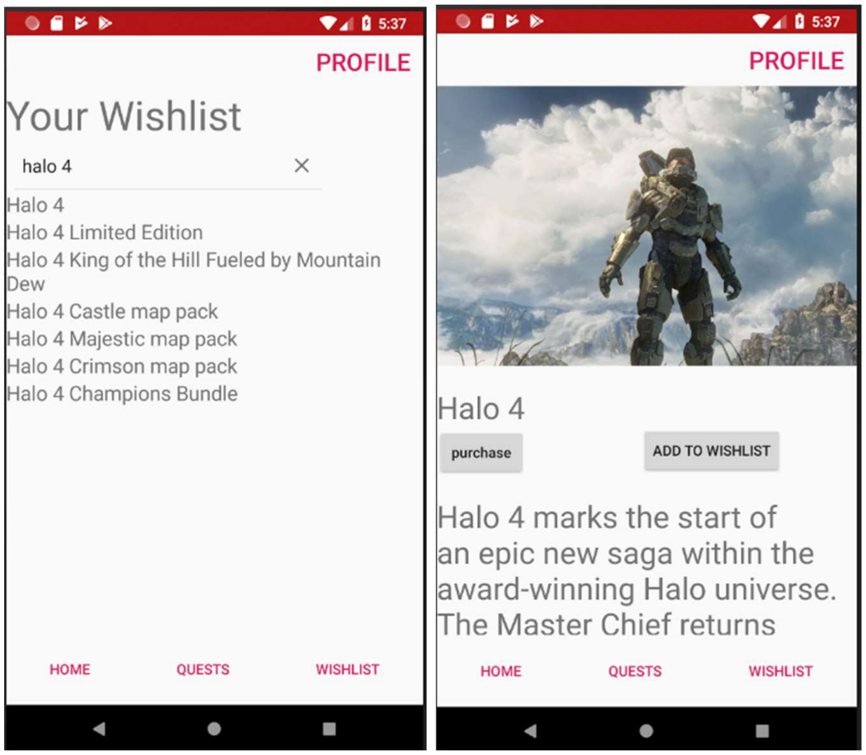
Fig. 8.1.3

Once the other user has completed the quest, you will receive a notification of its completion. Tap on the notification or navigate to the quests page to view the completed quest. Verify the quest was actually completed, to which you will be redirected to purchase the game once the function is implemented.

b. If you want to complete quests and earn games,

Follow the first three steps of the coach's manual above to register, sign in, and add a friend.

Now, click on the "wishlist" button at the bottom. Here, you can search for games you want to add to your wishlist that you want, and your coach can see and later buy for you as well. Go ahead and add a few to start.



Your Wishlist

halo 4

Halo 4
Halo 4 Limited Edition
Halo 4 King of the Hill Fueled by Mountain Dew
Halo 4 Castle map pack
Halo 4 Majestic map pack
Halo 4 Crimson map pack
Halo 4 Champions Bundle

Halo 4

Halo 4 marks the start of an epic new saga within the award-winning Halo universe. The Master Chief returns

Fig 8.1.4

Once you have a few games in your list, ask your coach to assign you a quest. Once you receive a notification that you have a quest, go to the app and click on the "quests" section on the bottom. Click on the quest to view what it means. Don't click "finished" just yet, you have to make sure you actually did the quest! The coach will confirm if it is done.

User manual will include transactions once the functionality is implemented in summer 2019.

8.2 FAQ

a. How do I create an account?

   When you first start the app, click the "register" button to take you to the register screen. This will allow you to input your username, password, and email to sign up for the App.

b. How do I log out?

   Click on the "profile" button, and then "logout". This will take you back to the login screen and log you out of the app.

c. Where do I connect with someone?

   On the profile screen, touch the "+" button and enter the username of the person you want to quest with.

d. How can I add a game to my wishlist?

On the wishlist tab, click on the textbox to search for a game. Once you have found it, select the game, and touch “add to wishlist”, or just press the “+” button to the right of the game’s info.

e. How do I assign a quest to someone?

Click on the bottom “quests” tab, and touch “assign new quest”. Enter the username of the person you want to issue a quest to.

8.3 User workflow

Word document linked below is confidential, requires password given to course instructor for viewing.

https://badcodev.github.io/BehaviorApp_6.3_Confidential.docx

9. Semester Review

9.1 Firebase implementation

Firebase, by default, utilizes email-based authentication instead of usernames. This is a design choice by google to keep authentication consistent with google and facebook sign in, which we overlooked when considering authentication methods. We retro-actively implemented usernames after a workaround was found using display names.

9.2 Planning development concurrently with school/work

Although our team has worked on a few projects together, they usually spanned at most a month and a half and had a smaller scope, which we were able to completely devote our time and effort into finishing. With other factors such as senior Corps responsibilities and only having 2 team members, we found it difficult to pace this project concurrent with our other activities and classes, and overestimated the amount of time available. Although we still plan on completing the MVP, most of the polish will be concluded during the summer.

9.3 Meeting with clients

Our clients have been in a similar situation as #2, and communication has been somewhat inconsistent as there are limited times where all 4 project stakeholders are available at the same time to meet. To remedy this issue, we decided to find times where development and Loran can meet, since he is the bridge between the vision Josh sees for this app and the reality of our development progress.

9.4 Design philosophy moving forward, Legal overhead

In our market research, we determined that similar projects have been implemented in individual schools and reward systems but seemed to have failed due to clunky UI, difficulty of use, and a variety of similar superficial factors that could have been corrected. In order for our product to be successful, we had to plan in detail the goals and challenges we would need to face while registering Odjin as an LLC. Unfortunately, this created a lot of overhead in beginning the project, as we could not start development until the LLC was registered and the IP could be claimed.

9.5 Android, API, and firebase learning curve

There was a steeper learning curve to delve deeper into android, as constant revision and fine tuning are needed. This increases the difficulty of the project, as tasks need to be optimized in runtime since the lifecycle state of the application is quite fragile. If an incorrect call is made at the incorrect time it causes the whole app to crash, and we found this stability to be an issue when implementing the API.

10. Future Directions and Development

The biggest lesson we can learn moving forward from the challenges highlighted above is effectively scheduling app development depending on the number of developers we have. Below are the features and steps we plan on implementing in the future after Spring 2019:

1. Extended functions, robustness

   Many of the functions implemented suffice for the most basic use of the application. Because there are many elements to juggle when developing an android app, we were unable to implement as many features to the quest building and profile pages as we hoped. The next step would be to extend the capability of the app in terms of quests and basic functions to provide a fleshed out user experience.
2. Transactions

   The core business model of our app requires the use of transactions through the codeswholesale API. We were unable to implement this this semester, so implementing this after ironing out the extensions listed above is key to creating an MVP for release.
3. Optimization

   Due to the heavy use of threading in android and limited processing power, optimization is needed to save both battery life and smooth user experience so the app runs smoothly and no busywaiting occurs.
4. UI/UX Review

   At this stage, we would need to review how our app displays the systems we have set up to the user. Especially for phones, UI/UX is extremely important, as our app will not be used or sell if this is not effectively executed. This includes seeking advising for colors to use, layout of app elements, consistency of the app experience, and assessing how easy it is to accomplish what a user seeks to do with the app.
5. Review for minors

   Since the primary market for our app are parents, we need to take a serious look at how our app handles information for minors. Explicit content must be filtered, and we must carefully choose how we display information to anybody under the age of 18 or risk severe losses in the future. This is a non-negotiable step, and legal counsel may be necessary when reviewing what laws surround applications used by children.

6. Security review

   One of the final steps would be to review how secure our app is. We have built it from the ground up with security in mind, so this should not pose a serious threat, but any networking capabilities and access to the database need to be re-reviewed before conducting user testing or submitting to the google play store.
7. User testing

   See section 6
8. Submission to google play store

   Google has relatively loose restrictions on what can be submitted, but stability is one of the biggest concerns in this step. Testing and review must be completed to the fullest extend before submission for the app to succeed.

11. Acknowledgements

Project Management

- Loran Hauserman
  loranhauserman@gmail.com

Behavioral Science subject matter expert

- Josh Hauserman
  hausermanj10@gmail.com

Student development team, project members

1. Lawrence Glick
   lawg97@vt.edu
2. Richard Patten
   richp@vt.edu

CS 4264 Instructor

- Prof. Edward Fox
  fox@vt.edu

12. Guides and References

The following references have been selected and briefly described due to their relevance to the project, as each reference is used in some portion of the app:

1. **Android Studio Documentation & Reference** -
   As of 5/7/2019.
   https://developer.android.com/docs
   https://developer.android.com/reference
   https://developer.android.com/guide
   a. Android Development fundamentals

    i. Overview on how to develop for android apps - https://developer.android.com/guide/components/fundamentals
    ii. How to construct app resources and layout in XML - https://developer.android.com/guide/topics/resources/providing-resources
    iii. Quality Assurance Guidelines - https://developer.android.com/docs/quality-guidelines/core-app-quality

b. Activities
    i. Introduction - https://developer.android.com/guide/topics/resources/providing-resources
    ii. Detail on an activity's lifecycle - https://developer.android.com/guide/components/activities/activity-lifecycle
    iii. Description of foreground/background tasks - https://developer.android.com/guide/components/activities/tasks-and-back-stack

c. Fragments
    i. Overview - https://developer.android.com/guide/components/fragments
    ii. Fragments and their relationship to UI flexibility - https://developer.android.com/training/basics/fragments/fragment-ui

d. UI
   https://developer.android.com/guide/topics/ui
    i. Layouts, Viewgroups, and interaction with the app - https://developer.android.com/guide/topics/ui/declaring-layout
    ii. Description of LinearLayout - https://developer.android.com/guide/topics/ui/layout/linear
    iii. Description of RelativeLayout - https://developer.android.com/guide/topics/ui/layout/relative
    iv. Customizing UI components & adapters - https://developer.android.com/guide/topics/ui/custom-components

   v. Style of app components (look and feel) - https://developer.android.com/guide/topics/ui/look-and-feel
   vi. Dialog text input - https://developer.android.com/guide/topics/ui/dialogs
   vii. Options menu - https://developer.android.com/guide/topics/ui/menus
   viii. Creating a search bar and interface - https://developer.android.com/guide/topics/search
   ix. Drawables & graphics - https://developer.android.com/guide/topics/graphics

e. Notifications - https://developer.android.com/guide/topics/ui/notifiers/notifications
f. Multithreading & its relation to UI events (important for implementing the API)
   i. When to use background tasks https://developer.android.com/guide/background
   ii. Threading (**IMPORTANT FOR API USE**) - https://developer.android.com/training/multiple-threads
   iii. Optimizing threads & background tasks - https://developer.android.com/topic/performance/background-optimization
   iv. Increasing performance through threading - https://developer.android.com/topic/performance/threads
   v. Performing network operations, necessary for API usage - https://developer.android.com/training/basics/network-ops
g. Handling user Input
   i. Responding to user input events - https://developer.android.com/guide/topics/ui/ui-events
   ii. Handling keyboard input, changing keyboard layout - https://developer.android.com/training/keyboard-input
h. Unit testing
   i. Overview - https://developer.android.com/training/testing
   ii. Device capability and flexibility - https://developer.android.com/docs/quality-guidelines/building-for-billions-device-capacity

        iii. Monitoring performance and finding bottlenecks - https://firebase.google.com/docs/perf-mon
        iv. Testing firebase - https://firebase.google.com/docs/test-lab
    i. App security
        i. Best practices, what to avoid - https://developer.android.com/topic/security/best-practices
        ii. Network security confirmation - https://developer.android.com/training/articles/security-config
        iii. Cryptography library - https://developer.android.com/guide/topics/security/cryptography

2. **Firebase**
   Current as of 5/7/2019.
    a. Android documentation - https://firebase.google.com/docs/reference/android/packages
    b. Setting up firebase - https://firebase.google.com/docs/android/setup
    c. Further detail on the relationship between android and firebase - https://firebase.google.com/docs/projects/learn-more
    d. Authentication
        i. Introduction to implementing authentication - https://firebase.google.com/docs/auth
        ii. How users are handled in an app - https://firebase.google.com/docs/auth/users
        iii. Retrieving/updating/sending user information - https://firebase.google.com/docs/auth/android/manage-users
        iv. Implementing password based authentication - https://firebase.google.com/docs/auth/android/password-auth
        v. Enabling email link authentication - https://firebase.google.com/docs/auth/android/email-link-auth
        vi. Adding google sign-in to the project - https://firebase.google.com/docs/auth/android/google-signin
        vii. Integrating google play games - https://firebase.google.com/docs/auth/android/play-games
    e. Real-time Database

   i. DB overview - https://firebase.google.com/docs/database
   ii. Implementing the database - https://firebase.google.com/docs/database/android/start
   iii. How to structure data from a higher level - https://firebase.google.com/docs/database/android/structure-data
   iv. Attaching listeners, reading and writing data to the DB - https://firebase.google.com/docs/database/android/read-and-write
   v. List / Object representations in the database - https://firebase.google.com/docs/database/android/lists-of-data
   vi. Offline persistence - https://firebase.google.com/docs/database/android/offline-capabilities
   vii. Database Security
      1. Understanding how firebase handles security rules - https://firebase.google.com/docs/database/security
      2. How to structure rules to secure data in firebase - https://firebase.google.com/docs/database/security/securing-data
      3. User - based security - https://firebase.google.com/docs/database/security/user-security
      4. Guide on resolving security issues with firebase - https://firebase.google.com/docs/database/security/resolve-insecurities
   viii. Automating backups - https://firebase.google.com/docs/database/backups
   ix. Storing files (eg. profile picture) on firebase - https://firebase.google.com/docs/storage/android/start

f. Analytics & Growth
   i. Machine learning based growth predictions - https://firebase.google.com/docs/predictions
   ii. Firebase crash analytics reporting - https://firebase.google.com/docs/crashlytics

iii. Attaching event-based analytics to the app - https://firebase.google.com/docs/analytics/android/start
iv. Implementing google ads - https://firebase.google.com/docs/ads

3. **API**

Current as of 5/7/2019

a. Documentation - Listed on confidential supplement
b. Supplemental API | used to retrieve game metadata while we received the API key - https://www.igdb.com/api, https://api-docs.igdb.com/