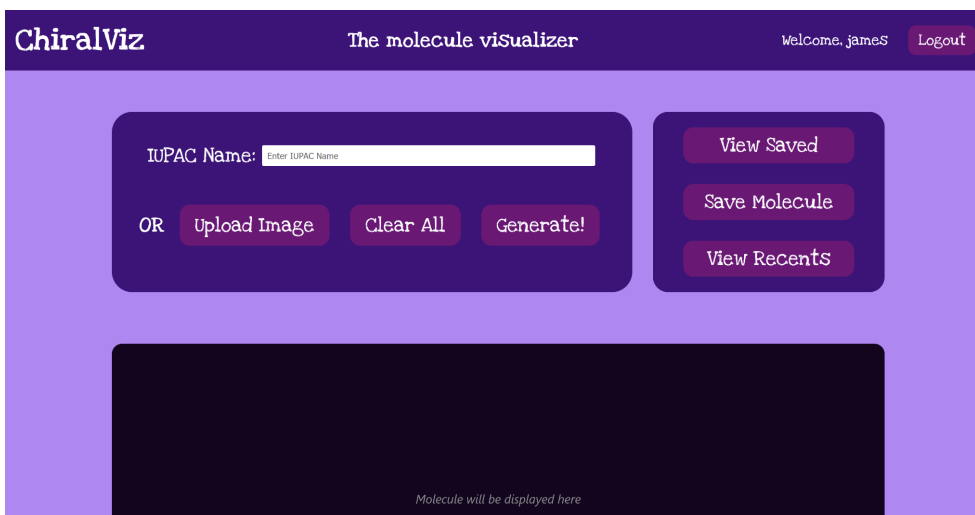
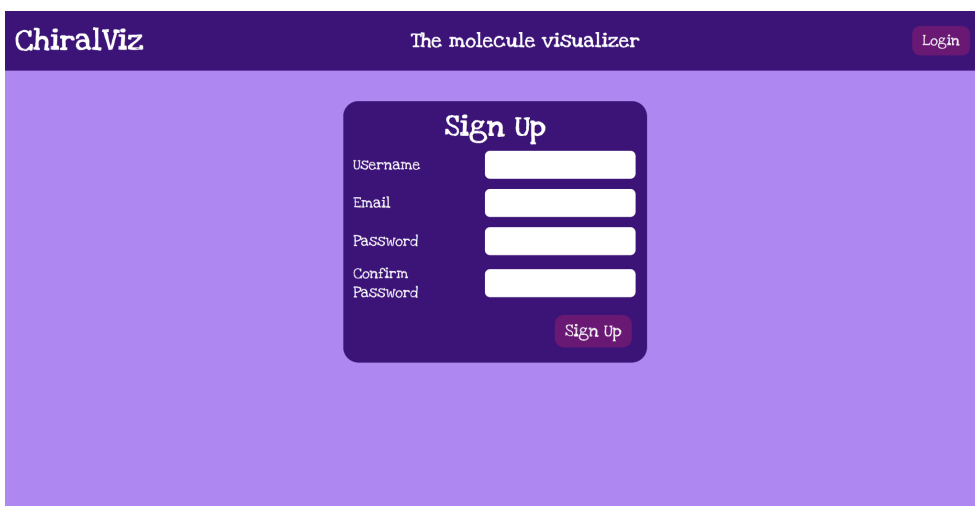
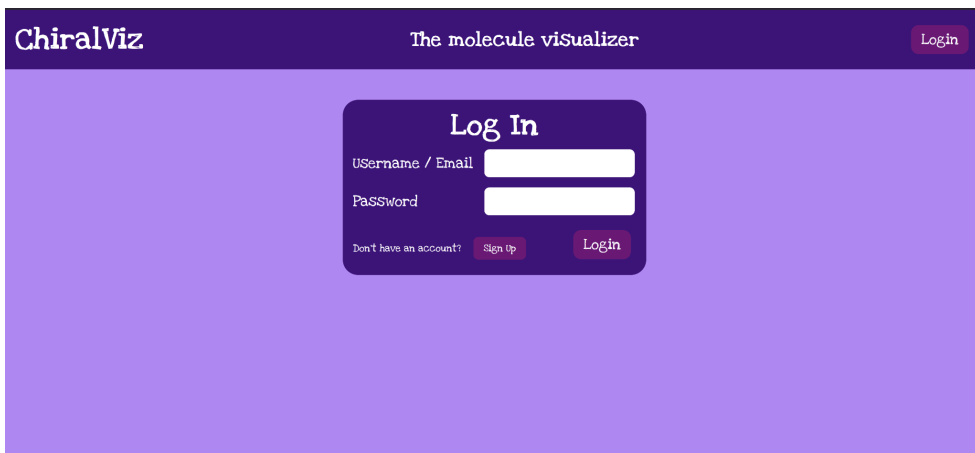


- Brief product description
 - If you were to describe your project to someone in an elevator, what would you say? Write a paragraph to explain what your project is all about.

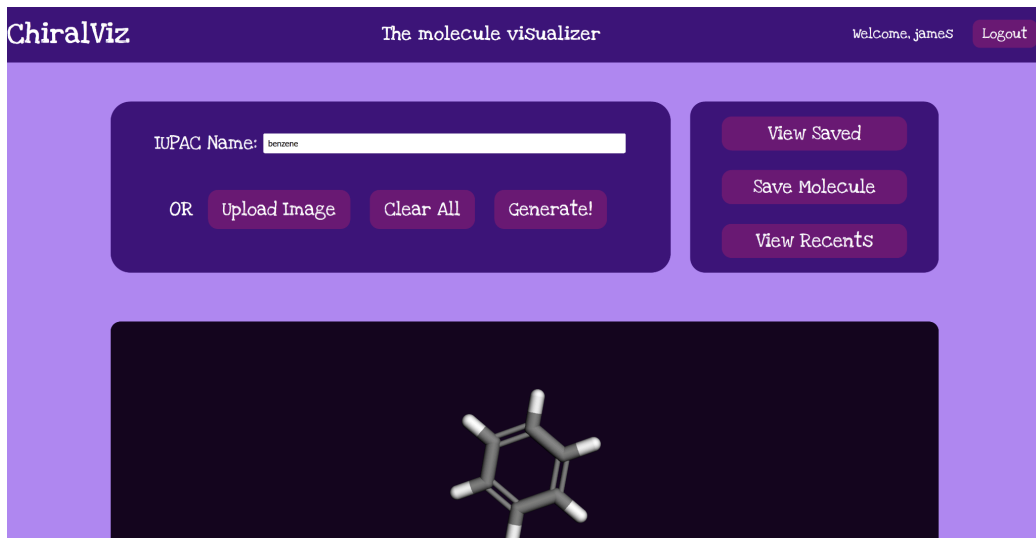
ChiralViz is an innovative web application designed to address challenges associated with understanding stereochemistry by converting traditional 2D molecular representations into intuitive 3D models. Students, researchers and professors can easily visualize complex molecular structures by inputting IUPAC names or via the transformation of molecular images bridging the gap between theoretical concepts and spatial intuition.

- Product functionalities
 - Using screen shots, showcase the functionalities of your product and walk a user through the experience of working or using your product.
 - i. User Authentication
 - Users can sign-up or login securely to access personalized lists.



ii. IUPAC Name Parsing and Visualization

- A user can input an IUPAC name to receive the relevant visualization.



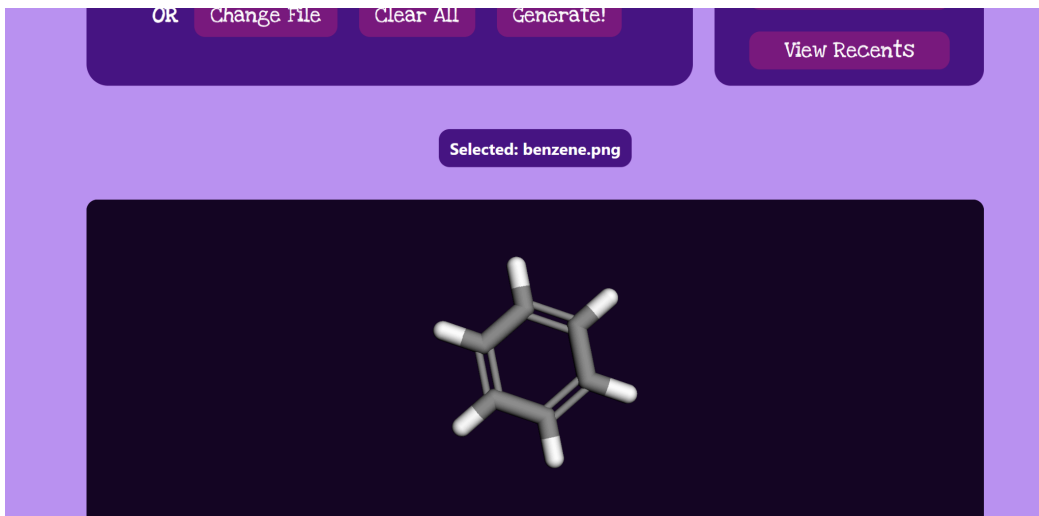
iii. Common Name Parsing and Visualization

- A user can input an IUPAC name to receive the relevant visualization.



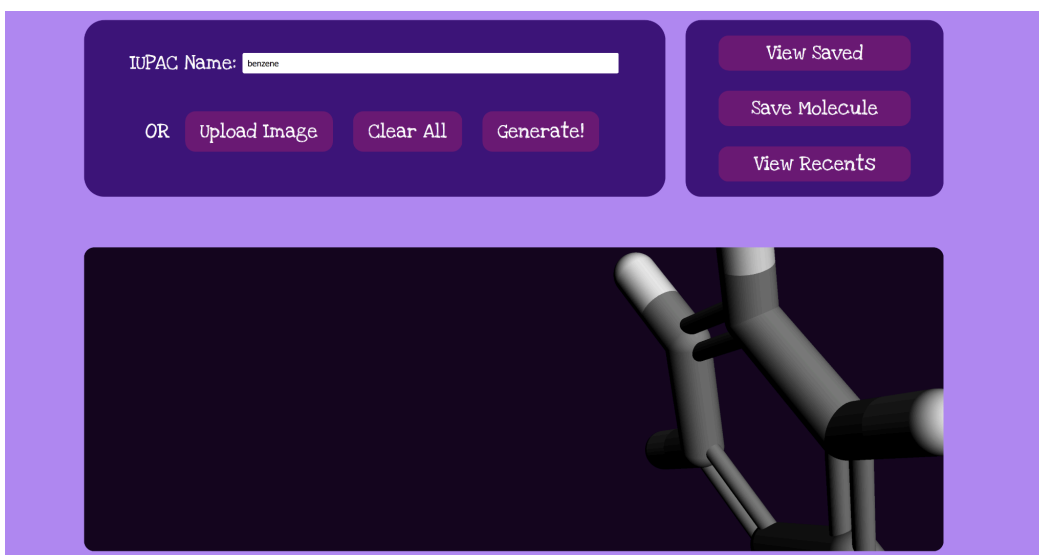
iv. 2D Image Recognition

- The system interprets uploaded 2D images to generate accurate 3D visualizations.



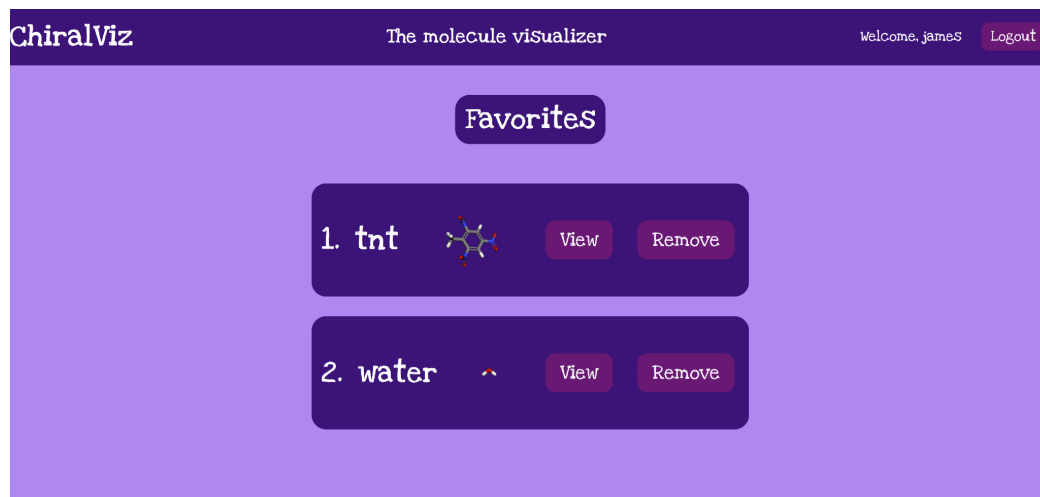
v. Molecule Visualization

- The generated 3D visualization is interactive allowing for rotation and exploration from multiple angles.

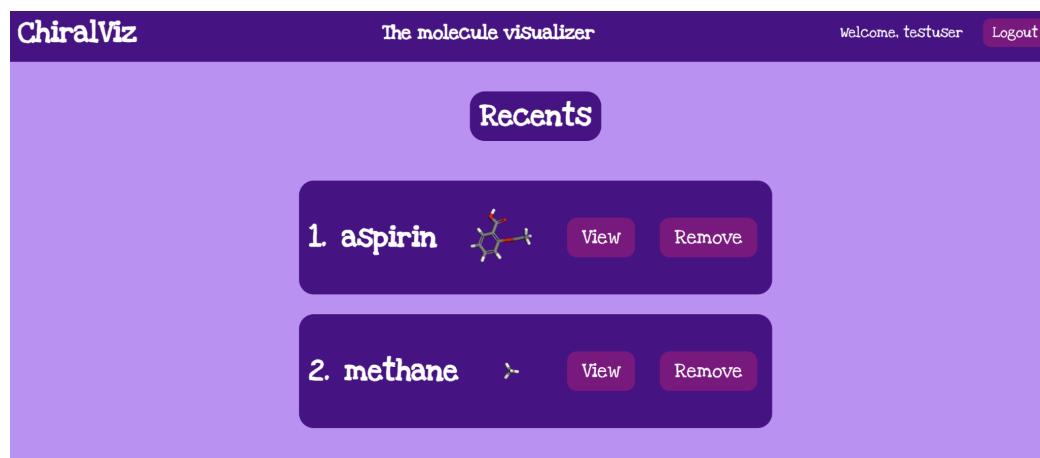


vi. User Data Management

- Once logged in a user can save IUPAC names to a favorites list and manage this list for easy access.

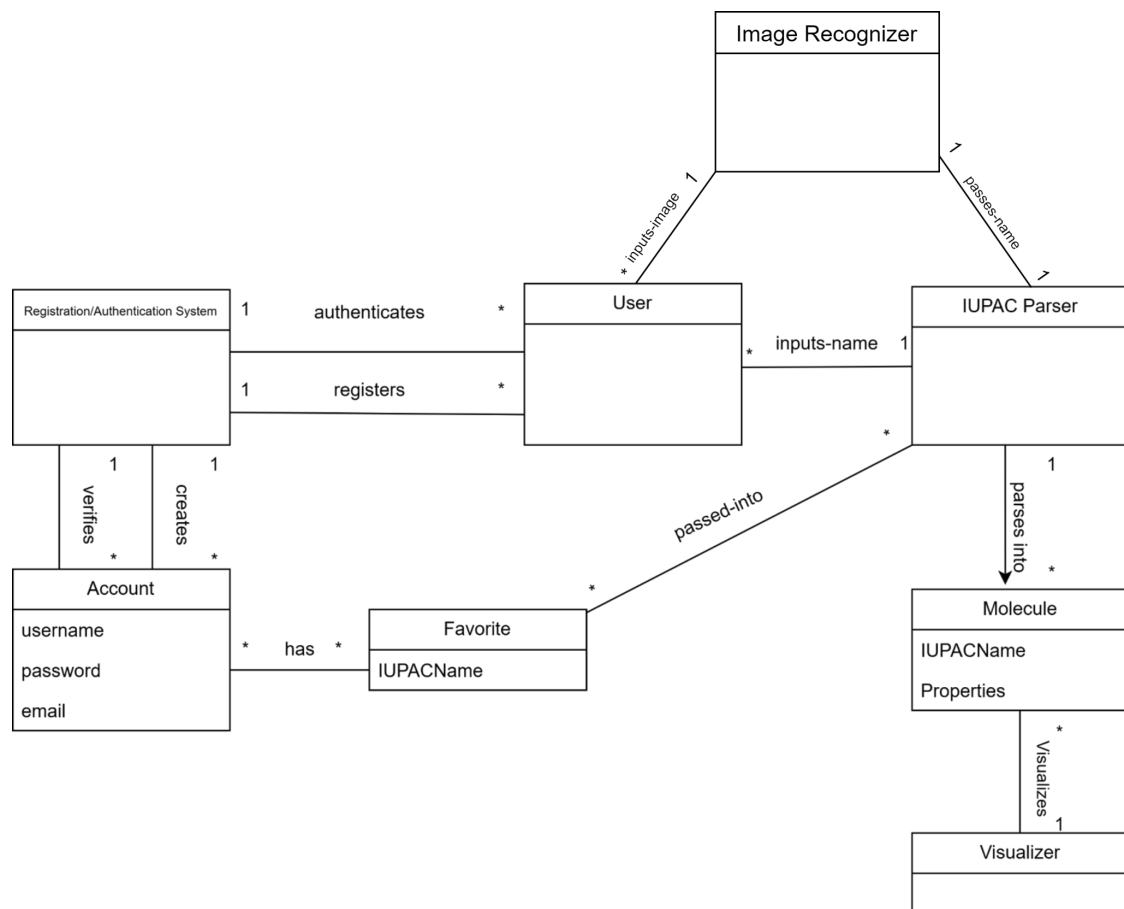


- The user is able to view their recent searches along with relevant visualizations.



- Design
 - Explain your design thoroughly. You can add diagrams of your domain model, interaction diagrams, and design class diagram. If you used algorithm, models, equations and parameters to implement the project, explain how they are used, how they are chosen, and the process you went through to utilize them for your purposes. Also explain what tools were adopted, and what is unique about your product and how it is different from similar products.

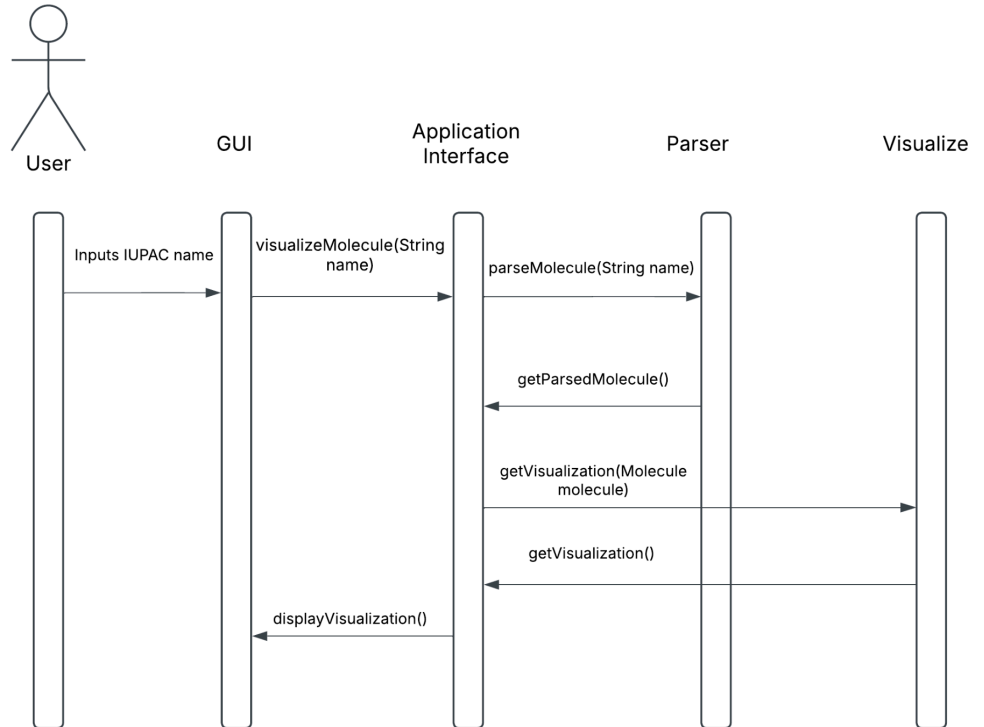
i.



- A new **user registers** an *account* with the *system*.
- The *registration system creates* an *account* for the new *user*.
- An unauthenticated *user authenticates* their credentials with the *authentication system*.
- The *authentication system verifies* that an *account* with these credentials exists.
- A *user inputs* an IUPAC name into the *parser*.
- A *user inputs* a 2D image into the *image recognizer*.
- The *image recognizer extracts* the IUPAC name from the image and **passes** it to the *parser*.
- The *parser parses* the properties of the IUPAC name into a *molecule*.
- A *molecule is visualized* by the *visualizer*.
- An *account has* favorite IUPAC names.
- A *favorite can be passed into* the *parser*.

ii.

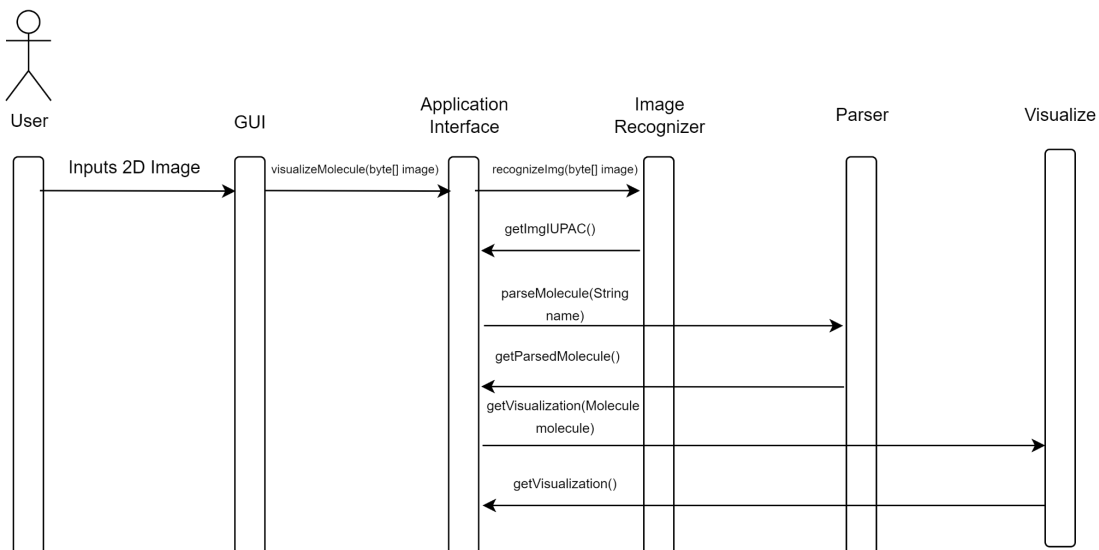
Use Case 1: IUPAC Name Parsing - Sequence Diagram



- The user inputs an IUPAC name to the GUI.
- The GUI requests a visualization of the IUPAC name from the Application Interface.
- The Application Interface requests the Parser to return a Molecule from the IUPAC name.
- The Parser returns the molecule.
- The application interface requests a visualization of the Molecule from the Visualizer.
- The Visualizer returns the visualization to the Application Interface.
- The visualization is displayed on the GUI.

iii.

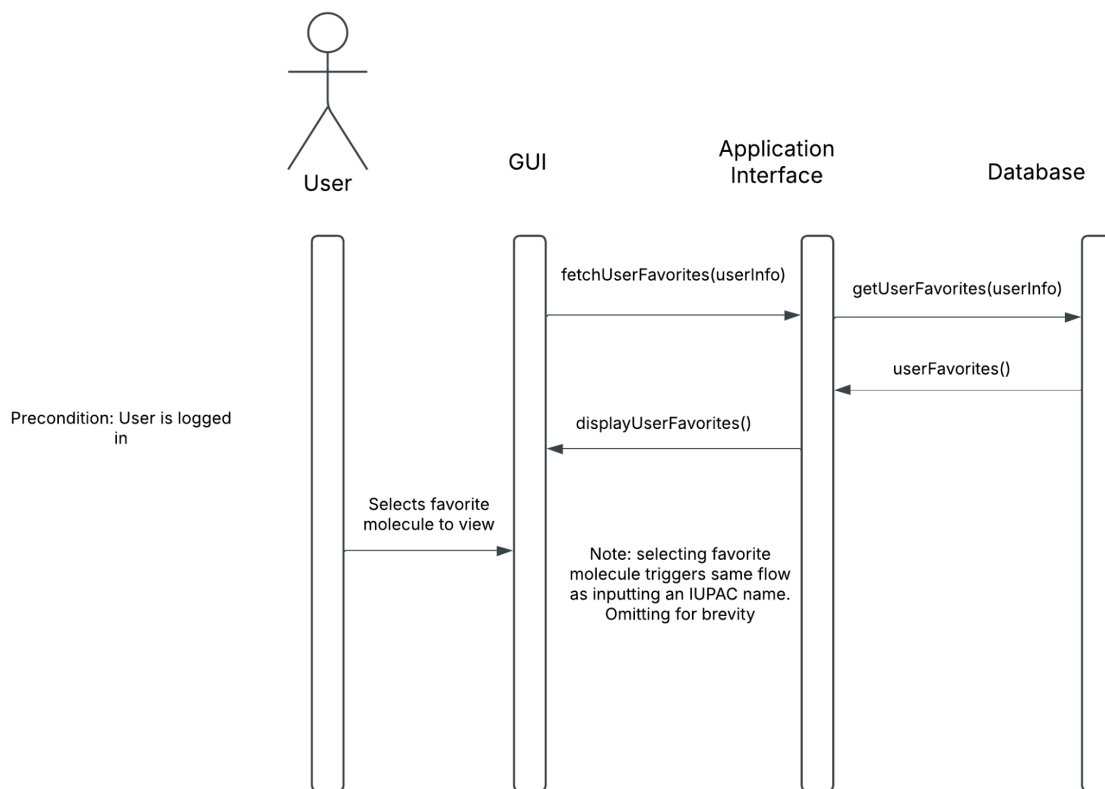
Use Case 1a: 2D Image Input - Sequence Diagram



- The user inputs a 2D image into the GUI.
- The GUI requests a 3D visualization of the image from the application interface.
- The application interface requests the IUPAC name of the image from the image recognizer.
- The image recognizer extracts the IUPAC name from the image and returns it to the Application Interface.
- The IUPAC name parsing flow is followed from here on out.

iv.

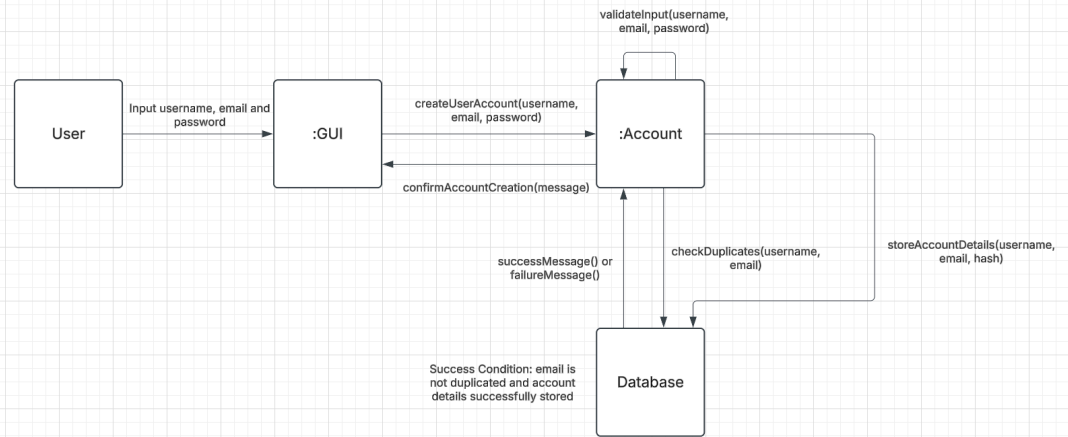
Use Case 4: User views a favorite molecule - Sequence Diagram



- The GUI requests the favorites of a logged-in User from the Application Interface.
- The Application Interface queries the Database for the User's favorites.
- The User's favorites are returned to the Application Interface.
- The User's favorites are displayed on the GUI.
- The User selects a molecule (IUPAC name) to view.
- The IUPAC name input flow is called.

v.

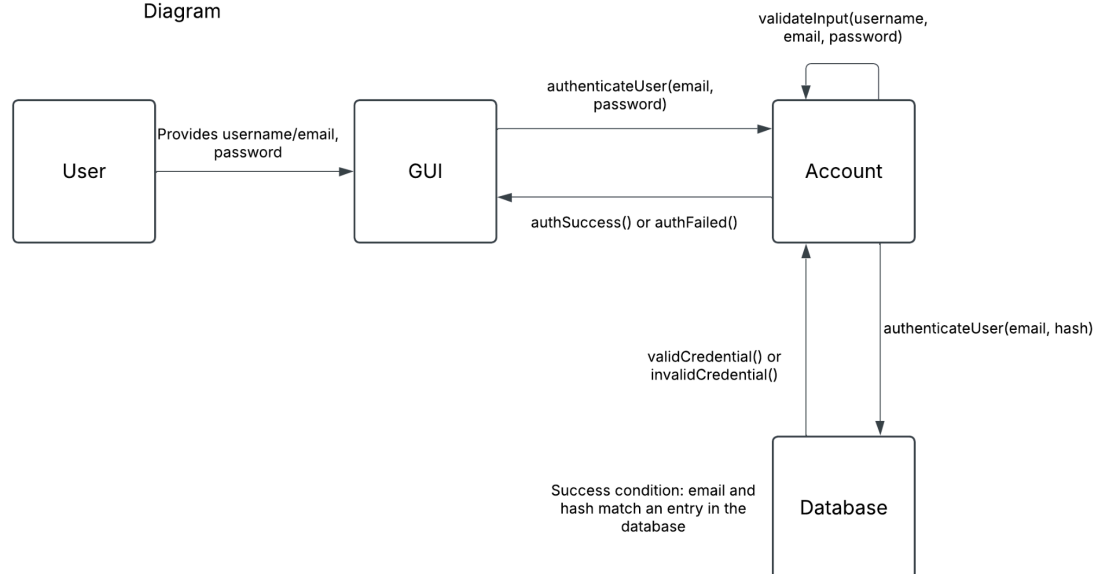
Use Case 2: User Signs Up - Collaboration Diagram



- A new user inputs username, email and password for account creation.
- The GUI calls a createUserAccount method with these parameters.
- If the input is valid and no duplicates exist in the database the account is successfully created and stored.
- The password is stored in an encrypted fashion.

vi.

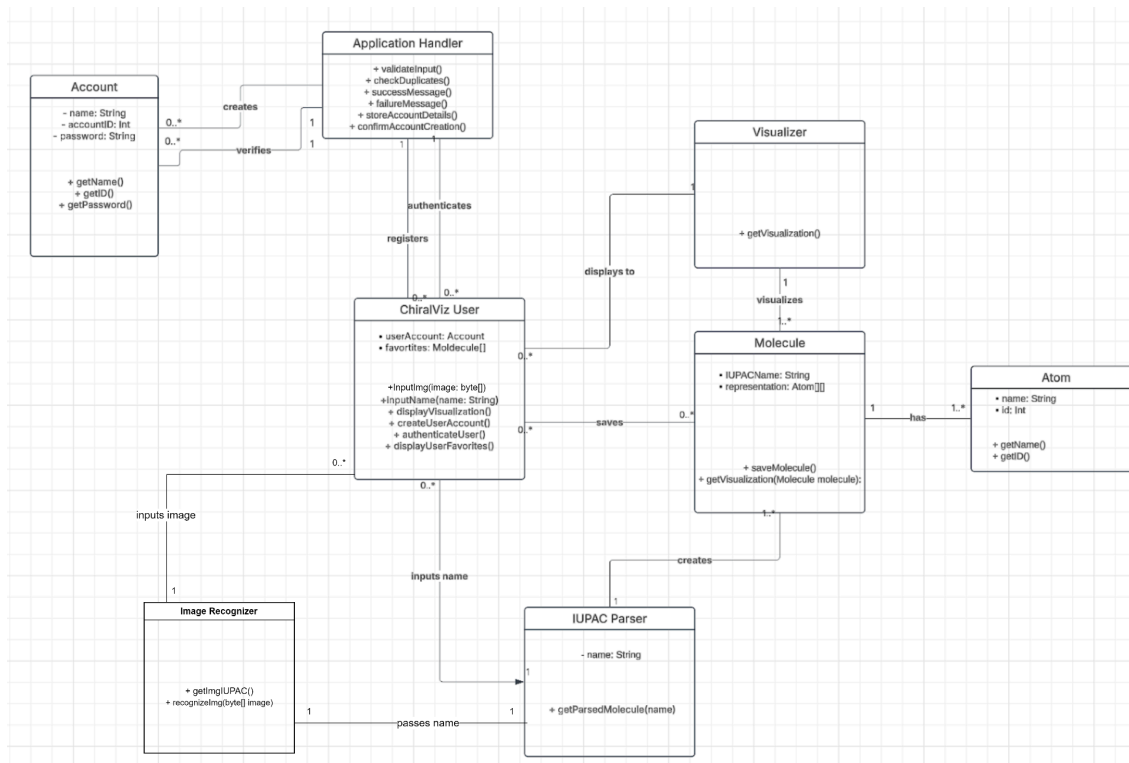
Use Case 3: User Logs In - Collaboration Diagram



- An unauthenticated user provides their credentials to the GUI.
- The GUI calls the authentication method.

- The username and password hash are validated and checked against the database.
- A success or failure response is produced depending on whether the credentials are valid.

vii.



•

Class	Purpose
Account	Stores data related to user authentication.
Application Handler	User registration and authentication logic.
ChiralViz User	A system user with molecule favorites
IUPAC Parser	Parses IUPAC name to molecule.
Image Recognizer	Extracts IUPAC name from 2D image.
Molecule	Represented parsed molecule structure and properties.

Visualizer	Renders a 3D visualization of a Molecule
Atom	An Atom in a molecule.

viii. Tools Used:

- OPSIN API: To aid in conversion of an IUPAC name to a SMILES code.
- RDKit: To generate an SDF molecule description from a SMILES code.
- OSRA: To aid in conversion of a 2D image to SMILES code.
- 3Dmol.js: Interactive 3D visualization of molecules.
- JWT (JSON Web Tokens): Authentication and session management.
- Docker: Application Containerization.
- Kubernetes: Application Deployment.
- MySQL: RDBMS (Database Layer).
- Python Flask: Middle Tier API (Logic Layer).
- Typescript: Frontend Dashboard (Presentation Layer).
- Pytest: Unit Testing
- GitHub: Version Control and SCM.
- Trello: Project Management.

ix. Uniqueness:

- Existing molecule visualizers are often desktop applications that require manual assembly of atoms and bonds whereas ChiralViz allows users to input an IUPAC name or upload a 2D image for instant visualization.
- Existing solutions require complex software installation on client machines, whereas ChiralViz is web-based and accessible from any browser. Further, the 3 tier architecture leaves room to extend this application into a mobile app as well.
- Unlike its competitors ChiralViz is completely Open Source. Which enables ease of adoption and expansion.

- Retrospection

- What went well

- The team hit key milestones such as IUPAC parsing early.
- The team acted quickly and was able to huddle together to decide alternatives when faced with blockers such as OPSIN only having 2D coordinate data.
- Adopting SCM tools like Git and project management tools like Trello kept the team efficient and conflict free.

- What could have been better

- During Sprint 1 the team relied on only a single solution (OPSIN) only to discover that it did not produce the expected data.
- Initial uncertainty around authentication mechanisms delayed backend integration.
- Members of the team who were only working on a specific layer of the application needed to locally install and run other interacting layers during early phases.
- Discuss what you have learned and how working on this product added to your skills.
 - External tools should be validated early in development.
 - Feasibility checks and analysis stories are a must.
 - Having a high level architecture to follow significantly simplified the development process.
 - Hands on experience with the entire lifecycle from design to deployment and containerization using docker and kubernetes was invaluable experience.
- Describe actions that should have been taken to avoid the problems encountered while working on the project.
 - A significant issue was over-dependence on a single external solution. To avoid this in the future the team could use:
 - Analysis stories to validate expected behavior and output.
 - Have multiple alternative solutions in mind.
 - A prototype phase prior to the actual implementation.
 - Setup contingency buffers during the sprint for research and rework tasks.
 - Deployment of application layers to dev environments in the cloud early-on would reduce the time wasted on local setup of layers that were not being worked on by a developer.
- Describe how could your product get better.
 - Autocompletion of most frequent searches.
 - Investigating and implementing caching solutions to increase speed and efficiency.
 - A mobile application would be even more intuitive and user-friendly.
 - Integration of molecule image sketching directly in the application.

- Saving the properties of every new molecule to the database to avoid repeatedly querying OPSIN would increase speed and efficiency.