CampusCloset Documentation

Mission Statement □

The market for second-hand, sustainable fashion is growing. Introducing your digital thrift store: CampusCloset is a platform for university students to swap clothing with other students, fostering a sustainable and economical approach to fashion. The app serves as a platform to reduce waste and promote the recycling of clothing, allowing users to interact with others as well as find recommendations for their closet.

User Cases

1. Philip (BU student):

- **Need**: Affordable and appropriate attire for a job interview
- **Solution**: Finds a suit through Campus Closet, saving time and money compared to traditional shopping.

2. Clara (MIT graduate fellow):

- **Need**: Wants to declutter her wardrobe and access a wider variety of clothing
- **Solution**: Waits for the expansion of Campus Closet to participate in swapping clothes.

3. Ethan (Freshman looking to define his college style)

- **Need**: Diverse and affordable clothing options to explore different styles
- **Solution**: Uses Campus Closet to try new styles affordably and sustainably by swapping clothes instead of buying new ones.

4. Mia (Environmental activist and student)

- **Need**: A platform that aligns with her values of sustainability
- **Solution**: Chooses Campus Closet to reduce fashion waste and promote clothing recycling among her peers

Our Sprint Timeline:

- 2/29 - First Pitch
- 3/1 - Brainstorming Sessions
- 3/20 - Development Started (Login and Authentication)
- 4/4 - Demo Presentation (Explore and Search features)
- 4/11 - Feedback Integration
- 4/18 - User Testing
- 4/25 - Final Touches
- 4/25 - Last Class and Public Release

Technical Documentation □

Authentication & Data Storage

- **Firebase Realtime Database** is used for secure user login, authentication and data storage, handling session management and user data efficiently. The database stores: "imageUri", "itemName", "listingId", "tags", "location" and "userEmail" fields with each item.

Location-Based Services

- **Google Geocaching API** and **Android Location Services** ensure that users can find and swap clothes within their local area, enhancing the convenience and reducing the carbon footprint associated with shipping. On the main explore page, we query Firebase by the location tag, and users are only displayed items from their "city." A location permission request pops up once per device, once it's accepted, the same device will always have location services enabled for the app.

Auto-Tagging of Uploaded Images

- **Imagga API** automatically tags uploaded images, simplifying the listing process for users and enhancing the searchability of items. We used Retrofit to manage network requests. "Garment", "clothing", and "apparel" are common tags the API recognizes so we simply remove them! Included 40% confidence cutoff.

Search and Discovery Features

- Users can perform detailed searches by tags or titles, utilizing optimized Firebase queries for fast and relevant results.

Tinder-Like Swiping Interface

- Implemented using `androidx.cardview.widget.CardView` and Glide, this feature allows users to quickly browse through clothing options in a fun and engaging way.

Challenges and Solutions □

Git Version Control

- **Challenges**: Numerous conflicts between commits and difficulties in managing branches.
- **Solutions**: Adopted more rigorous branch management strategies, utilized git stashing and rebasing to maintain a clean commit history.

API Integration

- **Challenges**: Initial inaccuracy in tagging from Imagga API, leading to irrelevant tags.
- **Solutions**: Adjusted the confidence thresholds for tags and implemented filters to exclude common but unhelpful tags.

Time Management and Debugging

- **Challenges**: Integrating features within the tight timeline of the academic semester proved challenging, compounded by extensive debugging sessions that were time-consuming.
- **Solutions**: Enhanced planning and time management were employed, along with more effective debugging practices like systematic logging and using debugging tools more proficiently.

User Interface Design

- **Challenges**: Creating an intuitive and appealing user interface that could operate efficiently across different devices and operating systems.
- **Solutions**: Iterative design improvements through usability testing ensured a more user-friendly interface.

Future Directions

- **Feature Enhancements**: Include secure payment options for premium features, develop a more sophisticated tagging algorithm, and create a personalized explore page.
- **User Experience**: Continuous improvement of the app's UX/UI to ensure smooth and intuitive user interactions.
- **Market Expansion**: Extend the service to include non-students and additional geographical locations.

Citations

- **The Noun Project**: Icons and other graphical assets.
- **Firebase**: For authentication and database solutions.
- **Google Developers**: Location APIs.
- **Imagga**: Image recognition services.
- **Stack Overflow**: Developer community for troubleshooting and learning