Timelock HiFi Prototype ReadMe

Isabelle L., Regina S., Sophia L., Anna V.

Tools Used

We built our prototype using React Native and Expo as the primary development frameworks

Operating Instructions

You can access the app on an android or iOS device using the following instructions.

1. Clone the repository from github and run *npm install* to install all dependencies
2. Install Expo Go on your mobile device
3. Run *npx expo start* to launch the development server
4. Scan the QR code with your mobile device to open the app or use this link.

App features

The app features a fully functional navigation system with:

- Home screen displaying current, upcoming, and previous timelocks
- Camera, audio, location, and voice memo integration for media capture
- Interactive timelock creation flow
- Detailed views for past and upcoming timelocks with swipe navigation
- Profile viewing and sharing capabilities

Core technical features include:

- Real-time camera functionality
- Real time location services and mapping features
- Gesture-based navigation (swipes between screens)
- Dynamic image galleries
- Display of intended social features (emoji reactions)
- Share functionality using native sharing

Limitations

Limitations of our app include:

- Limited backend integration. Most of our data is stored locally and resets on app restart.
- Limited to image-based media for pictures; video capture not yet implemented.
- Authentication system not implemented; uses mock user data.
- No real-time notifications system.
- No database connectivity for persistent data storage.
- Limited to a predefined set of emoji reactions.
- No AI integration for memento generation.

Wizard of Oz Items

To create the experience of a fully functional app, we simulated a few elements including:

- Push notifications are simulated through UI elements
- Time-based unlocking is simulated through navigation rather than actual time verification
- Location tracking is represented through static UI elements
- Social interactions (likes, reactions) don't persist between sessions
- Friend activity notifications are simulated with predetermined data
- Turn-taking system for media submission is simulated through UI flow rather than actual scheduling
- Media processing and filtering are represented through pre-set options

Hard-Coded Aspects

We also hardcoded a few elements of the app including:

- User profiles and friend lists are predefined in mock data
- Event details (titles, dates, times) are stored in static constants
- Image galleries for past events use predetermined sets of images
- Timelock durations and countdown timers use static values
- Event participation numbers and reaction counts are preset
- Media categories and types are predefined
- Friend groups and sharing options use static data
- Event locations and map data are predefined
- Emoji reactions and interaction counts use fixed values