

CS3354 Software Engineering
Final Project Deliverable 1

Digital Lifestyle Habit Tracker

Rhea Bhatia
Cameron Meyer
Mark Mondt
Jackson Nestelroad
Andrew Sylvester
Prathyusha Thiruvuri

CS 3354.001 — Final Project Proposal

Title

Digital Lifestyle Habit Tracker

Members

- Rhea Bhatia
- Sophia Maloney
- Cameron Meyer
- Mark Mondt
- Jackson Nestelroad
- Andrew Sylvester
- Prathyusha Thiruvuri

Description

We are developing a mobile application to track time spent in other applications with an emphasis on accountability. Users will be able to set limits on certain apps and monitor how they are using their device throughout the day with a variety of data visualization tools. Users can invite others to view and monitor their activity as well, allowing friends and family members to hold each other accountable in lowering screen time. Our application offers a new way to decrease screen time, because it's easier to seek goals together rather than alone.

High-level features:


- Track time on apps
- Limit time on apps or app categories
- Data visualization
- Invite others for accountability (mutual or not)

Reasoning

We find ourselves wasting so much time through social media and other distracting apps on our phones. Furthermore, it's even easier to neglect times of productivity for distractions when we are alone or when we know nobody else will know. Our goal is to provide a time-tracking application that will be more effective in decreasing screen time compared to other applications, and we believe increased accountability and transparency will achieve this goal.

Task Delegation

Tasks will be further delegated as more research is conducted and as the parts of our application solidify in the requirements gathering process. All team members will be responsible for contributing to each deliverable, including requirements, system design, and business estimates. Team communication will be conducted using GroupMe and in-person meetings will be scheduled as needed.



A loose delegation of tasks for the deliverables are as follows:

- Deliverable 1

- Domain Research (Mark, Jackson)
- Functional requirements (Andrew)
- Non-functional requirements (Sophia)
- Use case diagram (Rhea)
- Sequence diagrams (Cameron)
- Architectural design (Prathyusha)

- Deliverable 2

- Project Scheduling (Prathyusha, Mark)
- Cost, Effort, and Pricing Estimation (Rhea, Cameron)
- Test plan (Andrew)
- Competition research (Jackson)
- Conclusion (Sophia)
- Prototype/test code (all)
- References (all)
- Presentation slides (all)

Addressing Feedback on Project Proposal

There was no specific feedback given to us regarding our initial project proposal. The word “loose” is circled in our delegation of tasks because we only assigned tasks for the deliverables with limited knowledge on where each team member would work best. This flexibility has been removed for this deliverable, and the delegation of tasks is now much more strict.

Delegation of Tasks

- Andrew Sylvester – Software process model
- Everyone – Functional requirements
- Everyone – Non-functional requirements
- Mark Mondt – Use case diagrams
- Everyone – Sequence diagrams
- Jackson Nestelroad – Class diagram
- Cameron Meyer – Architectural design

GitHub URL

https://github.com/prathyushathiruvuri/3354-Digital_Lifestyle_Habit_Tracker

Software Process Model

The system will be developed using an iterative process model. Specifically, we will use the Scrum method to incorporate agile methodologies into the development of the application. This agile process will always be centered on the initial requirements established within this report and the project scope document. However, it is very possible that the wants and needs of potential users will differ from our initial requirements, so an iterative model will give us the flexibility to react to changing user requirements discovered in later iterations.

An iterative process model is ideal for our application because it produces business value earlier in the development lifecycle. Our team can create an application with the essential features and build and improve the application over time. Furthermore, this model allows our team to focus on customer research and testing earlier in the software lifecycle, preventing us from investing in big features, such as exporting data to a variety of formats, when they may provide no value for users at all. Project issues and changes can also be detected earlier. Altogether, an iterative process model provides us with more flexibility in the development of our application that will hopefully avoid major time and money losses in the later stages of implementation.

Functional Requirements

While there are a wide variety of functional requirements for this application, this report is restricted to a range between five and seven. Below are the core functional requirements for the application.

1. The application will track and save the amount of time spent on an allowed application from the moment it is opened to the moment it is either put in the background or it is closed.
2. The user can set daily or weekly time limits for specific applications, categories, or application groups.
3. The user can invite another user by email or phone number to overview and manage certain app usage data, time limits, and restrictions as an accountability partner.
4. The user can request application time limits and restrictions to be overridden by an associated accountability partner.
5. The application can present visualizations for data in the form of a table, bar chart, pie chart, line chart, or word cloud.
6. The user can export daily, weekly, monthly, yearly, or all-time data to a .csv, .xls, .pdf, .json, or .xml file.

Non-functional Requirements

1. Product Requirements

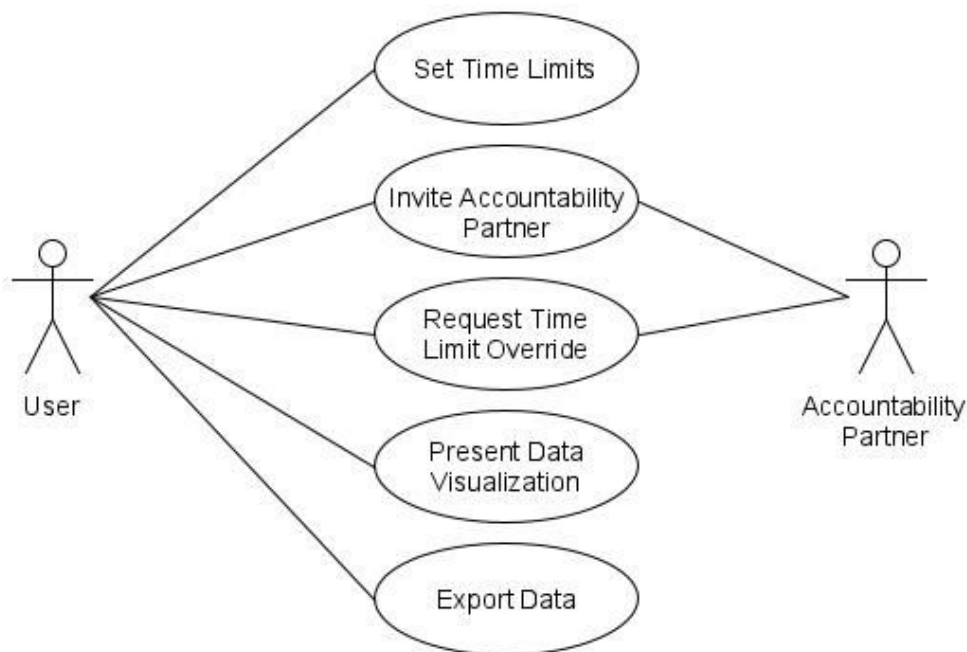
- a. **Usability Requirement** — All tutorials or help pages within the application will be no more than five steps. A simple tutorial will be available to the user on initial startup and in the help menu. Training time for the application should be no more than ten minutes.
- b. **Efficiency Requirements**
 - i. **Performance Requirement** — The application should have a minimal impact on the performance of the device it is running on. The difference between performing a set of tasks with the application running and performing the same tasks without the application running should be no more than 3%.
 - ii. **Space Requirement** — The application download size should not exceed 50 MBs.
- c. **Dependability Requirement** — The application shall succeed in retrieving the proper app usage data for 99.99% of user-initiated requests. The database server shall fulfill 99.99% of requests.
- d. **Security Requirement** — User passwords shall be stored in the database only after it is passed through a secure hashing function. All database requests shall be conducted over HTTPS.

2. Organizational Requirements

- a. **Environmental Requirement** — The application shall run on devices running one of the following operating systems: iOS 13 or higher; or Android 10 or higher.

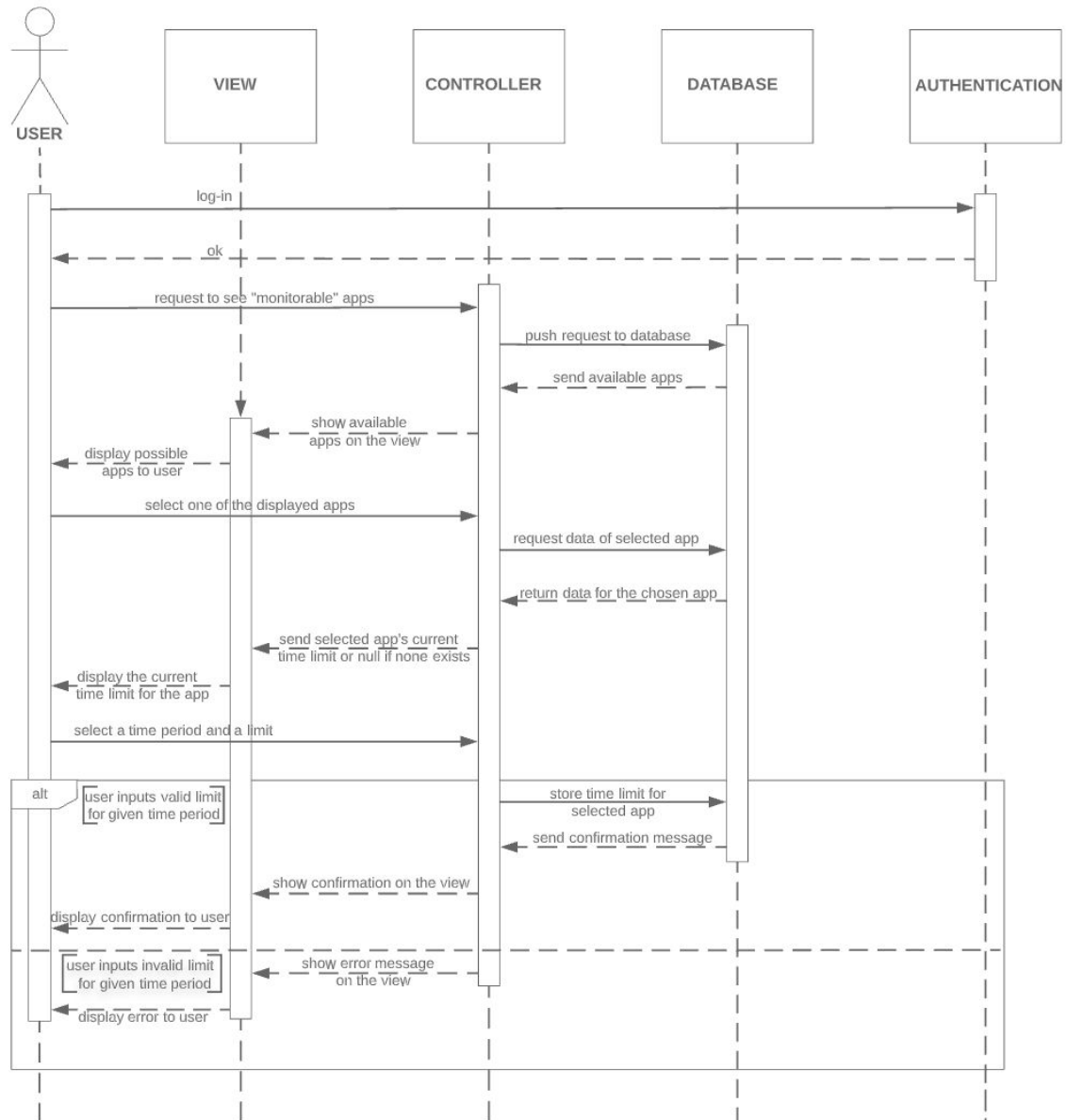
- b. **Operational Requirement** — The application shall be updated with every major release of the supported operating systems described above.
 - c. **Development Requirement** — The application shall be developed for cross-platform mobile devices using the Dart programming language and Flutter UI development kit.
3. **External Requirements**
- a. **Regulatory Requirement** — A user may request for all screen time data and account information to be deleted at any time. This requirement is in compliance with the European Union’s General Data Protection Regulation.
 - b. **Ethical Requirement** — No screen time data or personal information will be given to third parties of any kind. If data is to be sold to third parties in the future, explicit permission from the user is required to share their screen time data with external sources. No personal information of any kind will ever be shared with external third parties.
 - c. **Legislative Requirements**
 - i. **Accounting Requirement** — The total cost of development and maintenance for our app should not exceed \$50,000 per year.
 - ii. **Safety/Security Requirement** — The application shall provide an accessible link to the company’s privacy policy outlining what information is gathered, how it is shared with other parties, and how users can review and delete data. This requirement is in compliance with the California Online Privacy Protection Act of 2003.

Use Case Diagram

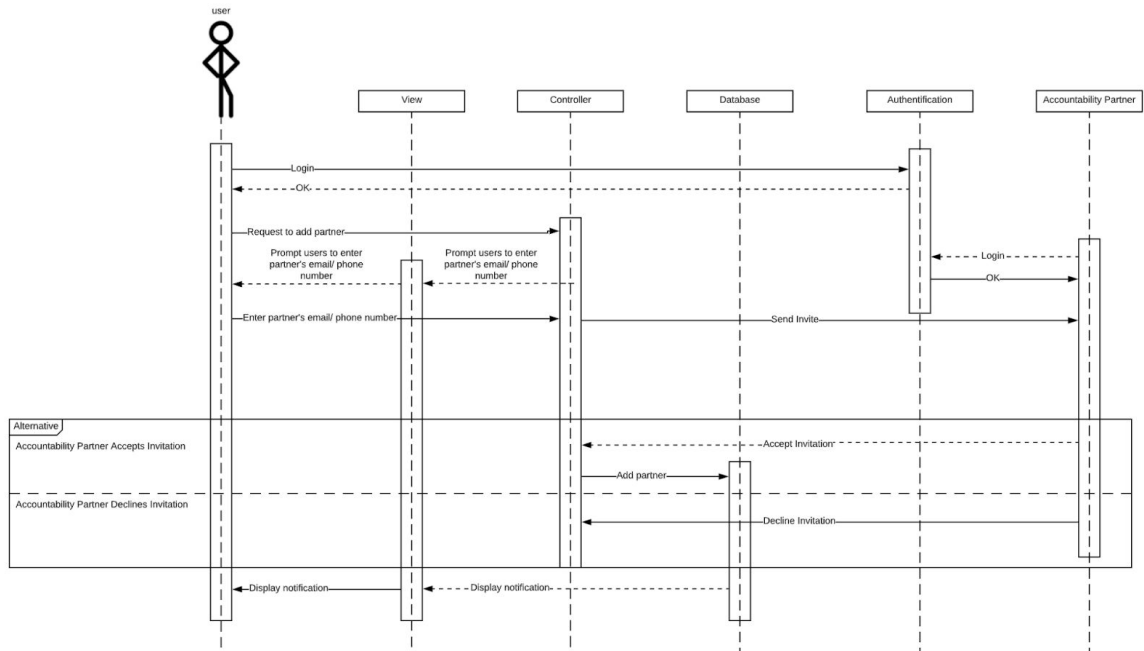


Sequence Diagrams

- Set Time Limits

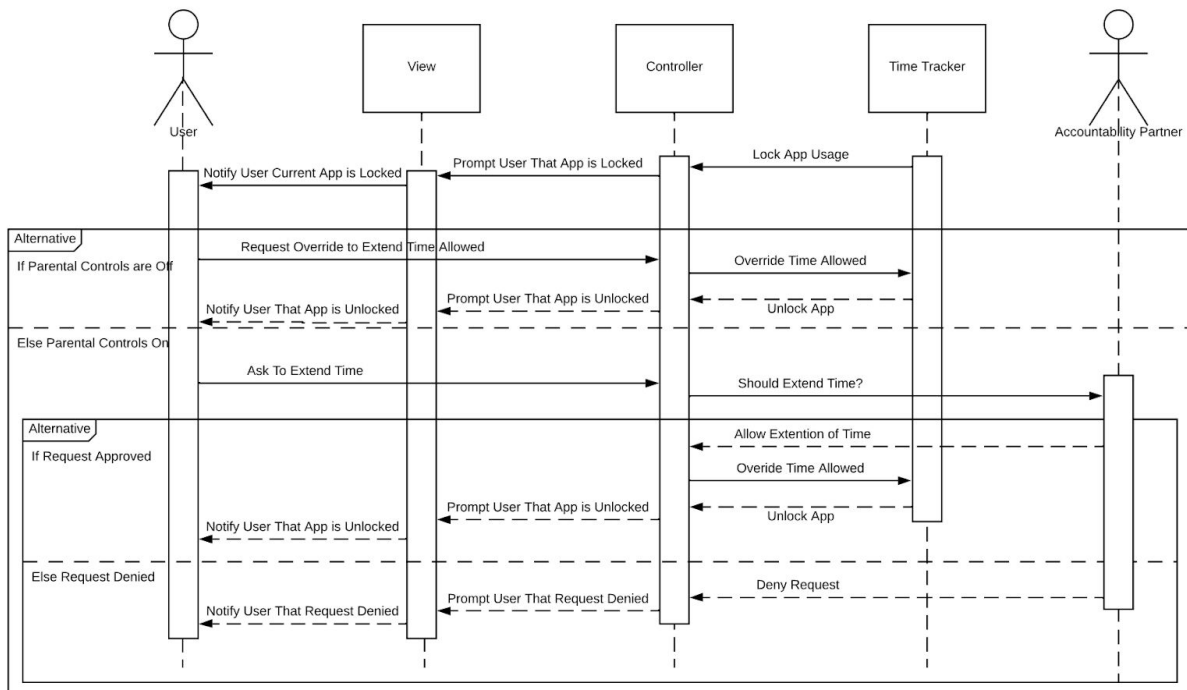


- Invite Accountability Partner

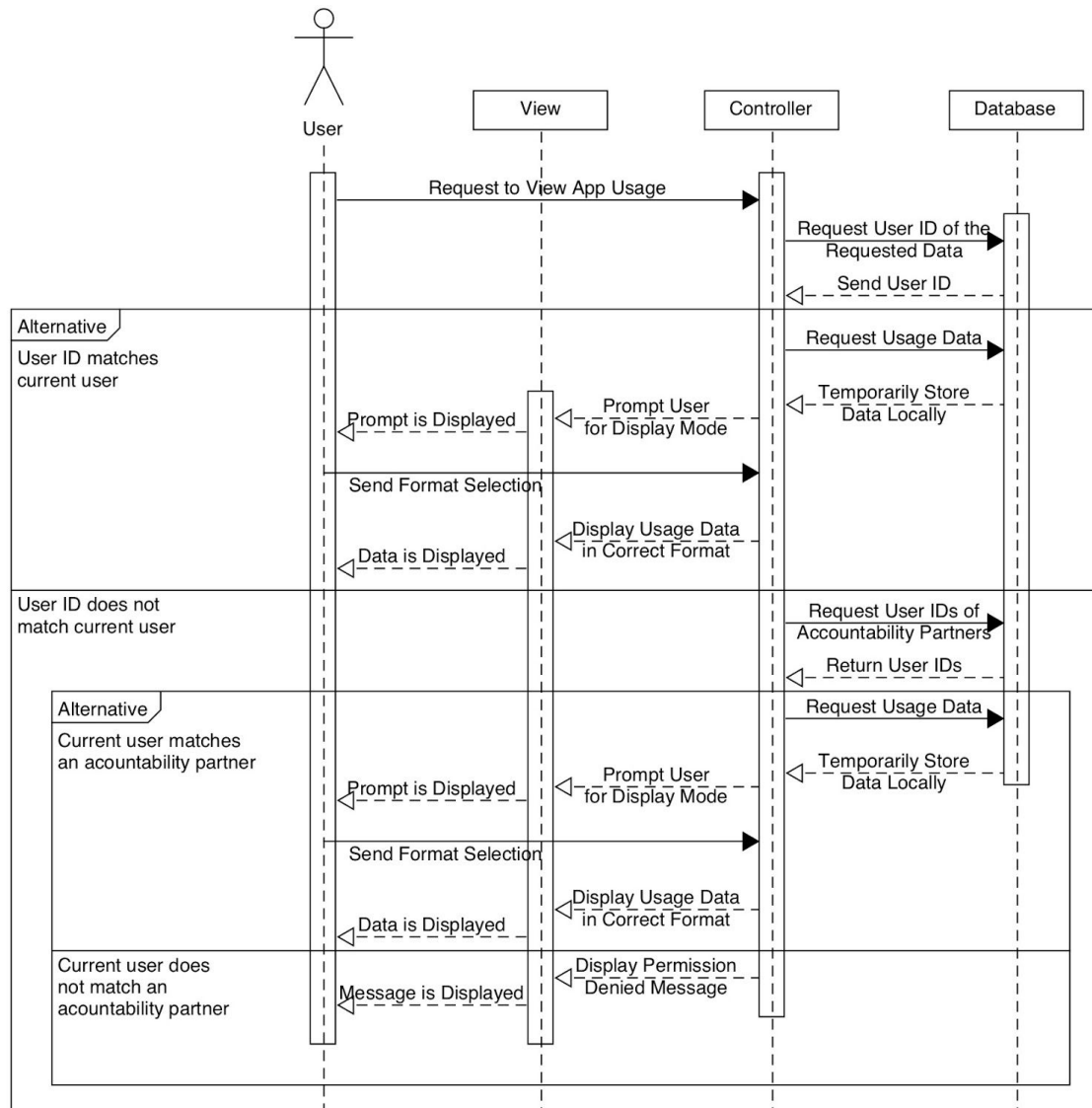


- Requesting Time Limit Override Sequence Diagram

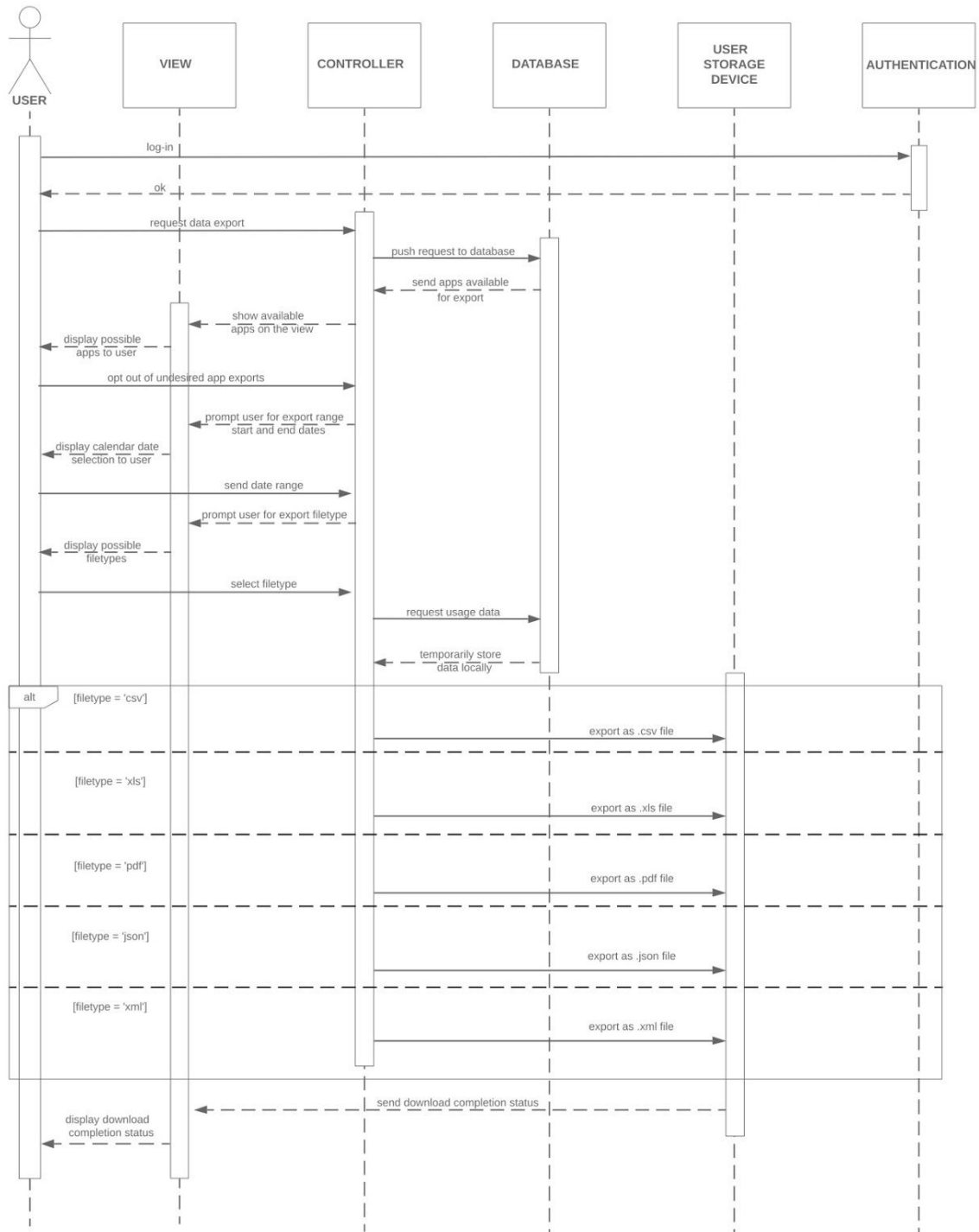
Request Time Limit Override



- Present Data Visualization



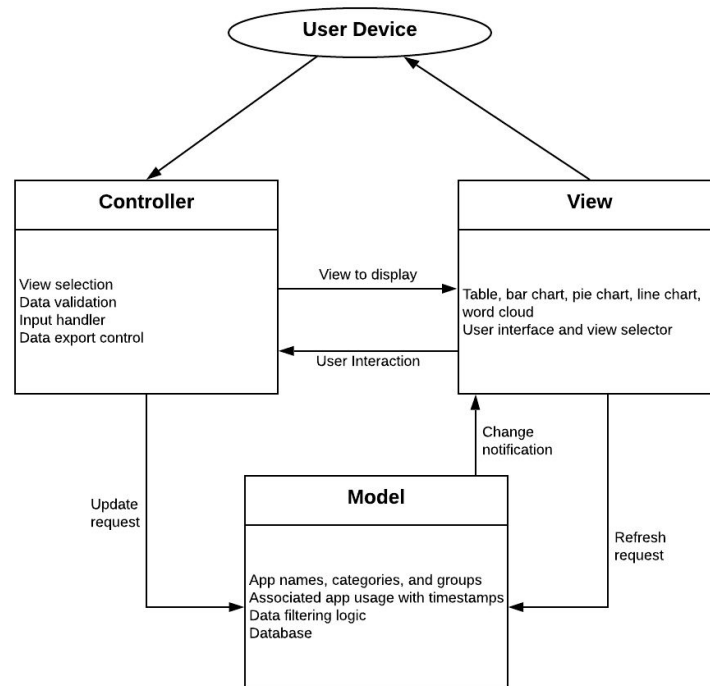
- Export Data



Class Diagram

Please see the included file named “Class Diagram.pdf” for the full UML class diagram.

Architectural Design



The application architecture follows the model-view-controller (MVC) pattern. The application is centered around the data collected from how the client uses their mobile device throughout the day. Our application also focuses on presenting this data to the user in a variety of ways, such as charts, graphs, and spreadsheets. Since all of these presentations, or views, will be acting on the same data, or model, the project fits very nicely with the model-view-controller architecture as the user is viewing and managing the way they use their mobile device.

For our application, the model component will be dynamically generated from the application database. The model and database will work together in the application to represent the user's screen time data.