# Team Plan

## Stakeholders

*List of stakeholders and their roles*

Our primary stakeholders are MIT students and community members. We require that users have an MIT email and verify it when registering in order to guarantee this. Students act as both lenders of items, and receivers of items on our application.

All four members of the team are experienced in web programming, and will code for the project to varying degrees; however to expedite organization and to provide structure in moving forward with implementing SnapBack, we have created roles for each member as described below. These roles have been chosen with each member's skill set and interests in mind.

**Jason**, Project Manager. Jason is SnapBack content architect, developing the pages, site structure, and investigating new features and studying use cases for the website. He will manage day-to-day operational aspects of projects including deadlines, targets, and allocation of resources, as well as modifying work plans as needed and diving in himself into SnapBack code to both create and review.

**Sarah**, Front-end Developer. Sarah will create the front end of the SnapBack website and will develop features and improve the user interface. She will utilize feedback from both team members and friends to improve learnability and efficiency.

**Stephanie**, Back-end engineer. Stephanie will handle setting up, programming and maintaining the database and server. She will also design server code with usage of gems and other third party tools as applicable.

**Paul**, Technical Director. Paul is responsible for overseeing all the technical operations of SnapBack and meeting scalability, performance, and reliability targets. He will develop a comprehensive suite of test cases including front-end user interface testing, data structure testing, as well as load testing on large number of requests to the server

## Resources

*List of computational, cost and time constraints*

Computational constraint is limited by the heroku cloud application platform. However, since SnapBack is not computationally expensive, and we don't expect extremely heavy traffic, the

constraint does not compromise SnapBack functionalities.

Since we are only MIT students and the application is on its first stage, we choose free option in heroku. Therefore, there is no cost for us to keep the application up and running.

The time constraint for completing the application is about 4 weeks. The final product needs to be delivered by Sunday, May 12.

# Tasks

*List of tasks, expected effort, allocation to team members*

**Features in MVP:**

- User registration/validation: allow users to sign up for the site with MIT email and send them a confirmation email after they have signed up to ensure that they are MIT students.
    - Includes: Sign up/in, verifying MIT email used, verification email
    - Time: 5 hours
    - Member: Stephanie
- Lend requests: allow users to submit a request asking for an item that they would like to borrow.
    - Includes: Create request for item, have other user verify
    - Time: 5 hours
    - Member: Paul, Jason
- Return requests: allow users to submit a request asking for an item back by a certain date. Allow user to specify how frequently they want to send reminders to the other person.
    - Includes: Create request for item return, include email reminders, have other user verify
    - Time: 5 hours
    - Member: Jason, Paul
- Email reminders
    - Includes: Change frequency of reminders for each return request
    - Time: 5 hours
    - Member: Sarah
- User Interface
    - Includes: Add AJAX requests for features (ex: user ratings), improve UI
    - Time: 10 hours
    - Member: Paul, Sarah, Stephanie, Jason

**Features Beyond MVP:**

- User Ratings: allow users to rate other users on a scale from 1 to 5, where 5 represents extremely trustworthy and 1 represents not trustworthy at all.
    - Includes: Rate users, update ratings, and view other users' ratings
    - Time: 5 hours

    - Member: Sarah, Paul
- Report requests: allow users to report items that they view as dangerous or illegal. Allow users to report other users as well.
    - Includes: Request items/users and include descriptions
    - Time: 5 hours
    - Member: Jason, Stephanie
- Tagging: allow users to tags requests for easier filtering and searching
    - Includes: Tag system and connections to requests
    - Time: 5 hours
    - Member: Paul, Sarah, Stephanie, Jason
- Images: add images to to requests for better user experience
    - Includes: Store images in database
    - Time: 3 hours
    - Member: Jason

Expected effort: each member is expected to share equal work and work on the project for at least 7 hours per week. Meetings are held at least once a week for at most 2 hours each. Each member will also look over code added to the project, so that they understand the current state of the application, how different parts are working together, and can review each others' code practices.

### *Calendar of intermediate and final milestones for tasks*

(class deadlines in black, ours in blue)

- Sunday April 21, 11:59pm: P 4.1 due
    - Design document
- Wednesday April 24th - some MVP implementation
    - Working sign up with email verification
    - Beginning to create basic lend and return requests
    - Beginning email reminders implementation
    - Basic tests for MVP features
- Sunday April 28, 11:59pm: P 4.2 due
    - MVP implementation
    - Programming: basic code only
- Monday April 29, 9:00am: P 4.2 demo due
    - MVP demo
- Wednesday May 1st
    - Ability to report items/users
    - Ability to rate users
    - Starting tags/images
    - Improved UI
- Sunday May 5, 11:59pm: P 4.3 due

    - Final product
    - Final design doc (update to reflect changes)
- Wednesday May 8, 9:00am: Project fair
    - Demo of final product
- Sunday May 12, 11:59pm: P 4.4 due
    - Final code (all parts, including unit tests)
    - Final design doc (updated to reflect changes)
    - Design Evaluation
    - Team Evaluation

# Risks

*Enumeration of expected risks and their mitigations*

**Too Many Features**
One risk while making this application is that we may have too many features (images, ratings, emails, etc.) planned to complete in the amount of time given. We ideally hope to finish all features, but if that's not possible, we'll implement features separately and in iterations so that the application will always be in a  working condition, and will still function without a feature.

**Slacking off**
Members of our team might get lazy and don't finish work by deadline. We will refer to the Team Contract, which clearly describes solutions to this type of problem.

# Minimum viable product

*Identification of minimum viable product for first release*

The MVP will include the ability to verify MIT accounts. Users can make a lend request for a specific item based on text input. They can also make a return request and specify a return date based on existing lend request. SnapBack will send a reminder email to the borrower if the return request is verified by both parties.

This first prototype will still be able to provide value to users in that users will be able to use SnapBack to make requests for items they need, items they want to return, and report items and users for invalid or illegal use. This will also provide a great opportunity for feedback from both team members and friends. We will make sure to have enough time in our process of designing and implementing to utilize user criticism to improve our application. We also believe this is a stepping stone to our final product in that all of the current features can be augmented and

improved in the final product.

### *Subset of features to be included*

We will include the following features for our first release of the product:

- User registration and validation
- Lend requests
- Return requests
- Email reminders and scheduling frequency

### *Issues postponed*

The MVP includes only basic functionalities. Security mitigations will be examined once we finish building the site structure because at that point we will have a clearer direction of security concerns. Beautiful user interface elements will also be included after the MVP is done.

We will also postpone more extraneous features like

- Report requests
- Item Tags/Searching
- Trustworthiness Ratings
- Images on items

### *Provides real value to users*

The MVP will allow users basic functionality of the application, which includes making lend and return requests. This is really the core use of the site. Our other features, such as a item images and tags, ratings, and reporting, can be postponed for our final product, as they aren't necessary for a basic idea of the application.

### *Provides opportunity for feedback*

Since the MVP includes basic functionalities of the application, it gives people who evaluate our application a good idea of what SnapBack does. The MVP will have enough features for them to provide feedback on things to add or change. Moreover, there are several possible extensions to be added to the MVP, so there is always room for improvement based on this feedback. It will let us know if how we should add the items in our 'issues postponed' section.

### *On path to full product*

Team: Starfleet

We will include the following additional features for the full product:

1. Trustworthiness rating
2. Tag for items for search feature
3. Images for items in lend requests
4. Report requests
5. Notifications for items you need to return