**SW Engineering CSC 648/848**

**HeapOverload - h0**

**Milestone 1**

**Section 01 Team 4**

**Team Members:**

**9/28/2022**

## Executive Summary:

Heap Overload is a way for those seeking a higher education to pose questions, review answers and connect with other scholars. Heap Overload or project h0 as it is called internally is a community based message posting application that will help organize and answer questions at a high level. Users can post questions and have them answered by other qualified persons in their respective fields. Questions will be vetted by how helpful they are to other users and will have a rating system to sort replies by the quality of answers. The messages will be organized by a tag system which will allow for general as well as more specific information to be organized efficiently. The tag system will also allow for users to find similar questions in a more time effective manner. Users will also be able to create communities of any size by connecting with other users to grow their respective networks. Heap Overload differs from other similar applications by its superior organizational systems. For many other applications it can be hard to find what you are looking for, but with Heap Overload finding relevant topics will be simple.

## Personas and User Stories: (**Name** persona *story*)

**Sachit lama** is a twenty year-old student at Contra Costa College. He lives with his roommate and is majoring in Bio Tech. He has a dog, and loves to hike and read books. He works part time and doesn’t have much savings. He uses web apps that don't have premium features. He’s been using the Internet for more than 10 years. Since his hobby is reading books, he wants something where he can ask questions about the books or to post reviews to like-minded users. He has used different websites and has tried to look at specific questions but it was a bad experience for him and wasn’t able to find what he was looking for. He has since stopped surfing the web as much and has resorted to asking Facebook and college friends his questions. *Sachit needs to be able to navigate to the pages that he's interested in, and see the posts that are relevant to him. Additionally he might also need to add a new post when he wants to ask a question, or reply to another post.*

**Shawn White** is a twenty-three year old male that lives in Cupertino, California, and is a student at the University of San Francisco studying Computer Science as an undergraduate. In his free time he likes to play video games and surf the web to learn more about computer science. When Shawn encounters a problem that he needs help with, he goes and looks up the question and browses relative questions that relate to what he asked. When he can’t find an answer that works for him he posts a question, and waits for replies to come. Sometimes while he is browsing other questions, he comes across a question that he knows the answer to and checks to see if any responses have been made, and will add his own answer to the list. If there are already good answers he will upvote the comment and continue browsing. *Shawn needs to be able to search for questions that he has, and have quick access to other relevant questions. He needs to have the answers easy to find, and when he doesn’t find his answer right away, he needs to be able to add new questions. Additionally, when he is just browsing he needs to see if posts have been answered or not, so that if he knows the answer he knows that someone else is waiting.*

**Lucy Lee** is a twenty-five year old girl at the University of Berkley working on their phD in Computer Science. She likes to read books, watch romance movies, and paint flowers. She works as an assistant teaching undergrad students. She is an experienced programmer, and wants to help others who are struggling. She likes to ask challenging questions to encourage others to think in new ways, and promote discussion among the other users in the replies. In addition to this she also enjoys browsing categories that she is familiar with and answering as many questions as she can to help people who are just getting started. One of her favorite ways of explanation is creating short videos to share. *Lucy needs to be able to navigate into categories of questions that she is familiar with and be able to answer questions that she finds. She also needs to be able to have her videos displayed in the replies to make it easier to see, rather than having to visit an external site. While she is reading questions she also goes and reads the replies and up votes on good comments and down votes on bad comments to help moderate replies. Since Lucy has been doing this for a long time, and has had her account verified, her replies will get pinned to the top of the comment section.*

## Data Definitions: (Define main terms and data structures)

Document for users:

Unique user id,

Login name,

Hashed password,

User email,

Friends [List of other user ids],

Posts [List of post ids]

Document for posts

Unique Post Id,

Title,

Author,

Date,

Replies [list of replies containing user id, reply, like count]

Document with tags and related posts

Tag1[list of post ids under this tag],

Tag2[list of post ids under this tag]

## Functional Requirements:

*Priority: 1 - ASAP; 2 - Important; 3 - When we have time*

| **ID** | **Title** | **Description** | **Priority** | **Story** |
| --- | --- | --- | --- | --- |
| 001 | Add Post | Users will need to be able to add a post in the category that they specify. | 1 | All |
| 002 | Reply to Post | Users will need to be able to reply to a post when they find it. | 1 | All |
| 003 | Search for Post | Users will need to be able to find a specific post they are looking for, either by tag or content. | 1 | All |
| 004 | Edit Post / Replies | Users will need to be able to update their posts and replies. | 1 | ~ |
| 005 | Vote on a Post | Users will need to be able to vote whether a reply was good or bad. | 2 | Lucy / Shawn |
| 006 | Delete Post / Reply | Users will need to be able to delete their post or reply when they want to. | 1 | ~ |
| 007 | User Accounts | Users will need to be able to make their own account. This will keep records of their posts, replies, and connections | 1 | All |
| 008 | Embedded Video Player | Users should be able to share videos and have them play on our application and not direct users to other websites. | 3 | Lucy |
| 009 | Verified Users | Some users should be able to apply to become a verified user, to show that they know what they are talking about and their replies and posts are shown closer to the top. | 3 | Lucy |
| 010 | Notifications | Users should be able to change their preferences on what notifications they get and how they get them. | 2 | ~ |
| 011 | Live streaming of categories | Verified users should be able to stream lectures or other informational information to a category. | 3 | ~ |
| 012 | Level of Urgency | Posters should be able to specify how urgently they need an answer to be solved. | 3 | ~ |

## Non-Functional Requirements: (Performance, storage space, usability, security, availability, fault tolerance)

Fast Responses with database requests (Show loading screen)

User can post as much as they want, no post limits, however we might run into some issues with a really large user base, and would need to migrate database to account for more storage

Make things function as you think as they should, color code buttons

Secure login information and hash sensitive information so that sensitive data is not stored in a raw form

Accessible from computer or mobile devices via the internet (rescaling for mobile devices)

Show default screen, send error message, time based response test

## Competitive Analysis: (3-4 features that make use stand out)

Tags for posts, this will all for more or less specificity, and allow users to post in a topic that relates to the post.

Live Streaming, lectures for people looking to learn from people who are verified users.

Level of urgency to show how fast or slow they need an answer to their question, and also sort posts by urgency

Verified Users by checking a users post history and rating their impact on the community, then invite them to become a verified user, showing their responses higher.

## High-level System Requirements:

|  | Technology | Version Number |
| --- | --- | --- |
| Server Host | Google Compute Engine 2vCPU 2GB Ram |  |
| Operating System | Debian | 11 (bullseye) |
| Database | MongoDB |  |
| Web Server | Caddy |  |
| Server-Side Language | Node.js |  |
| Web App Framework | Express |  |
| Front-End | React |  |
| Additional Technologies |  |  |
| IDE | VSCode |  |

## Team:

| Name | Role |
| --- | --- |
| Adam Garcia | Team Lead |
| Bishow Bhattarai | Scrum Master |
| Briana Sze | Front-End Master |
| Jasmeet Singh Bal | Front-End Master |
| Ming Chen | Back-End Master |
| Sakar Pokhrel | Github Master |

## Checklist: (Answer DONE or ON TRACK or ISSUE)

* Team found a time slot to meet outside of the class **DONE**
* Scrum Master shares meeting minutes with everyone after each meeting. **DONE**
* Github master chosen **DONE**
* Everyone sets up their local development environment from the team’s git repo. **ON TRACK**
* Team decided and agreed together on using the listed SW tools and deployment server **DONE**
* Team ready and able to use the chosen back/front-end frameworks.
  + For each technology (front/back-end/DB/cloud) , the team decides who will lead the study of each technology and what will be output of the (feasibility) study within one month. **DONE**
  + If you list a detailed study plan for this, earn extra points! **ON TRACK**
* Team lead ensured that all team members read the final M1 and agree/understand it before submission **ON TRACK**