# Small group project self-assessment

**Team name**: *Dromedary*

The table below contains a list of epics mentioned in the assignment. For each epic that the team produced features for, add a brief explanation outlining how the user can access and use it. You have been provided with some code that incorporates some of the features you need (though you can extend it). This self-assessment will be used to help the marker figure out how to access your system, so try to keep it clear and concise.

Do review the marking scheme for the small group project to understand how the project is assessed. You are reminded that the assignment is **not** to try and implement every single epic.

|  |  |
| --- | --- |
| **Epic** | **Implementation** |
| Epic 1  The web application should have user authentication and authorisation mechanisms. Users should be able to create accounts, log in, and manage their profiles. | *The features in this cell were provided in the scaffolding project.*   * *A home screen gives users the option to sign-up or login (URL/)* * *Users without account, who are not logged in, can sign up as new users (URL/sign\_up)* * *Users with an account can log in (URL/log\_in)* * *Logged in users can log out (URL/log\_out)* * *Logged in users can change their password (URL/password)* * *Logged in users can change their profile (URL/profile)* * *After completing an account related task, logged in users are redirected to a currently blank dashboard (URL/dashboard)* * *If a not-logged in user tries to access pages requiring login, they are redirected to the login page, and then to the page they were after following successful login.* * *A menu is available on all pages while users are logged in.* |
| No further implementation |
| Epic 2  Users can form teams, and invite team members to their team. | * Once a user is logged in they will see a side-bar containing a button allowing them to create a new team * The sidebar also displays all the teams the user is already a part of * Once the user clicks the button they are taken to a form (URL /create\_team/) where they can make a team by specifying Team name, description, additional members and invitation message (if new members are to be invited) * Submitting this form successfully sends invitations to other members and creates the teams * Invitations appear in a user’s notification box (URL *view\_invitation*) where they can accept and decline them * A team dashboard allows team owners to edit the team (e.g. add members etc.)(URL /edit\_team/) and delete the team if necessary. * On the team dashboard there is a button any user that is part of a team can use to leave the team |
| Epic 3  Each user can create tasks, assign tasks to other team members, and set due dates. | * A task must have a team to be created, so users can only create tasks on team dashboards * On each team dashboard there is a New Task button * Clicking on the button takes the user to a form to create a new task (/create\_task/) * The form allows users to input a task title, description, due date, team, team member to assign the task to, task completion status and priority * Submitting the form with valid data allows for the task to be seen on the correct dashboards * Users can also edit tasks using the Edit button on each task in the dashboard (URL /edit\_task/). * Through the edit tasks button, users can also delete tasks |
| Epic 4  Users provide a dashboard to view and manage assigned tasks, and team tasks. | * There are 2 different types of dashboard * A user’s personal dashboard that displays all the user’s current tasks (URL /dashboard/) * The team dashboard that displays all the tasks for a team the user is in (URL /team/team\_id/) * The dashboards displays the tasks as task cards * Each task card has an Edit button so that the user can edit that task (URL /edit\_task/) |
| Epic 5  Enable users to search, order and filter tasks. These could be based on name, completion status, priority, due date ranges, assigned developer, or team/project. | * All dashboards have filters allowing tasks to sorted based on the different attributes they were saved with * Tasks can be filtered by words contained in their title, their current completion status, priority, who they were assigned to and their due date * IN the personal dashboard they can also be filtered by team |
| Epic 6  Add a priority system for tasks, allowing users to assign priority levels and filter tasks based on priority. Include automated reminders for high-priority tasks nearing their due dates. | * The task form contains a priority field that can be set between 1-5. Users can choose what they believe the priority of each task is * Users can filter based on priority |
| Epic 7  Introduce task dependencies, enabling users to link tasks and establish dependencies between them. Ensure that the application adjusts task timelines based on dependencies. | Not implemented |
| Epic 8  Integrate a time tracking feature for tasks, allowing users to log time spent on each task. Provide summary reports of time spent on tasks over different periods. | Not implemented |
| Epic 9  Implement an activity log that records user actions within the application, providing an audit trail for accountability. | Not implemented |
| Epic 10  Add a notification system (within the constraints of a Django) to alert users about task assignments and upcoming due dates. | * The URL /view\_invitation/ is essentially the notification system * Users are notified when they have been asked to join a team * They can accept or reject team invites from the notification * The notifications are shown with an inbox icon in the menu bar, so users can always see their notifications |
| Epic 11  Introduce gamification elements to motivate users, such as achievement badges, leaderboards, or a points system based on task completion. | Not implemented |