

Project Plan

In this part of our documentation we'll go over our project plan for Elicomp, the application where users can create challenges, compete in them and compare rankings.



Elicomp logo

During the first 2 weeks of our internship, after receiving the problem we had to solve, we did research and thought about what features the application should have. This part contains an introduction talking more in-depth about the application and internship, but also our MOSCOW and use cases.

Introduction

Team

Our team consists of 2 students from Thomas More Hoge school:

- Seppe Meeus, Application Development
- Emre Ekici, Artificial Intelligence

The Client



Elision logo

Elision is an IT company which operates under the Cronos group. They offer SAP customer experience solutions to a wide range of customers. They have worked with clients such as Kréfel and Action. We did our internship at their branch in Hasselt, Belgium.

The Goal

Our internship goal is very broad in a sense that we were not asked to do anything specifically aside from building an application to promote fitness and healthy competition between the employees of Elision. The only technical requirement was that we use Java for the backend as our mentor is most knowledgeable in Java.

Given that our project task left room for interpretation and creativity we began to brainstorm ideas and solutions we could implement for our application: Elicomp. This took place during the first 2 weeks.

After a few sessions we decided that we wanted build an application that will:

- Allow users to keep track of their physical activities
- Include a team system in which groups of users can participate against each other
- Have challenges which will be competitions which users or teams can join to work towards a goal
- Utilize a streak system for users who wish to set personal challenges for their own physical fitness goals
- Implement a reward system to entice users to participate in challenges and perform more activities
- Integrate external applications such as Strava so users don't have to manually log each activity they perform

Sprint planning

- Throughout the duration of the internship we will be working in an agile way with sprints of 2 week durations.
- At the end of every sprint, a sprint demo will take place where we show our progress to the internship mentor.
- Based on the feedback from our mentor we will have a sprint retrospective and plan out our next sprint.

Reporting

The progress of the project can be followed through:

- Daily standups: Every day we will have a standup where we go over what we did the previous day and any obstacles we encountered.
- Sprint Demos: At the end of every sprint cycle we will hold a sprint demo where we show off the current progress of the project.
- Jira: The planning for the current sprint with story points, also has the backlog and previous sprints.
- Weekly reports: Each of us will send out an email every Friday that will contain what we've worked on for each day of that week.
- Confluence: The documentation for the project as a whole will be written on confluence for Elision and exported as PDF's for Thomas More.

Project Requirements Analysis

MoSCoW

Must have	Should have	Could have	Won't have
Leaderboard system	Notification system	Pet system	GPS location tracking
Challenge system		AI assistant	
Reward system		Recreational challenges	
Hosting		Tutorial	
Streak system			

Must have:

- Leaderboard system:
 - Will display the scores for individuals and teams
 - Challenges will have separate leaderboards to track progress within challenges themselves
- Challenge system:
 - Periodic challenges will be hosted by the app/company itself in which users can earn experience points through individual or team based performances
 - Users will be able to host their own challenges open to other users in friendly competition
- Reward system:
 - Users will be able to level up and earn badges to show off their accomplishments and skill
- Hosting:
 - The application will be hosted for presentation after development is finished.
 - The research into how the hosting will be accomplished can be found in.
- Streak system:
 - Will allow Users to create Streaks for themselves. These Streaks will help Users to be more consistent and earn experience points.

Should have:

- Notification system:
 - The application should send out notifications for different scenarios such as:
 - To congratulate them on certain achievements and milestones
 - To warn them when their streak is at risk of ending
 - To notify them that of new challenges including the details of the challenge

Could have:

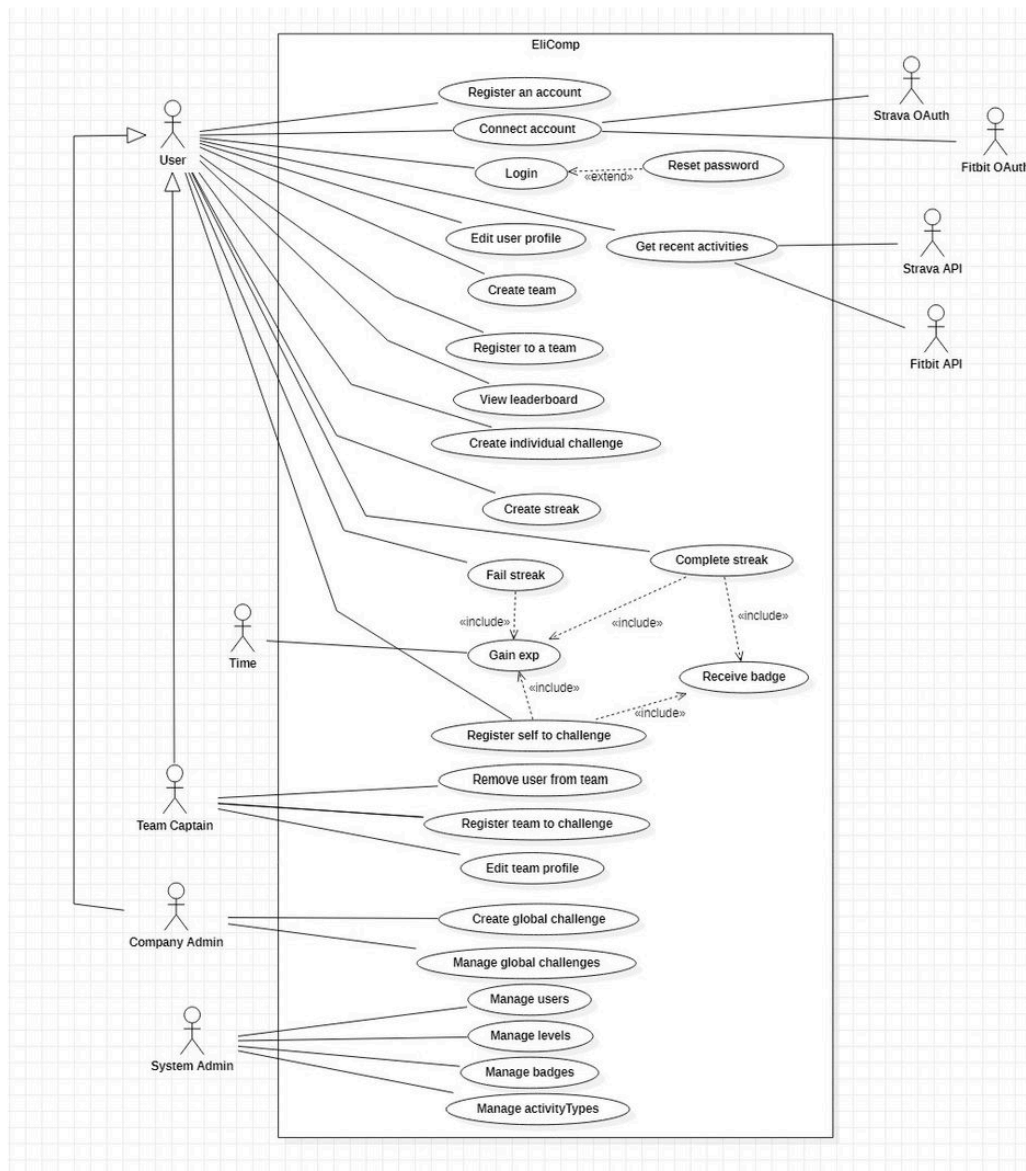
- Pet system:
 - The user could select a pet upon their registration
 - The pet system could serve as a continuation of the reward system and include features not limited to the following:
 - Cosmetics which could be earned depending on player level and progression
 - AI model so that users can interact with their pets

- AI assistant:
 - The AI assistant could provide the users with assistance regarding the following:
 - Questions about the usage of the app
 - Information about the challenge and reward systems
 - Training recommendations for the different activity types
- Recreational challenges:
 - The users would also be able to host competitions and track points for recreational activities such as darts, billiards, board games, video games etc.
- Separation of admins:
 - The way we'll go about admins is an admin can create a company wide challenge, but also manage users, levels and badges. Splitting this up in a company and system admin would be better for scalability.
- Tutorial;
 - A tutorial to walk users through the usage of the application.

Won't have:

- GPS location tracking:
 - As we will be building this app with data gathered from Strava and Fitbit, we believe the added value in creating a tracking system to follow users progress is minimal.

Use cases



Use case diagram

Use Case Descriptions

Register an Account

- **Description:** This use case allows a new user to create a new account in the system
- **Actors:** User
- **Preconditions:** None
- **Basic Flow:**
 - User navigates to the registration page
 - User provides necessary information such as username, email, and password
 - User submits the registration form

- **Postconditions:** User account is created in the system

Connect Account to Strava or Fitbit

- **Description:** This use case enables a user to link their system account with their Strava or Fitbit account
- **Actors:** User
- **Preconditions:** User is logged in to their system account
- **Basic Flow:**
 - User accesses the account settings or profile page
 - User selects the option to connect their account with Strava or Fitbit
 - User provides necessary details for OAuth
- **Postconditions:** User's system account is linked with their Strava or Fitbit account

Login

- **Description:** This use case allows a registered user to log in to their system account
- **Actors:** User
- **Preconditions:** User has a registered account
- **Basic Flow:**
 - User navigates to the login page
 - User enters their username/email and password
 - User submits the login form
- **Postconditions:** User is successfully authenticated and gains access to their account

Reset Password

- **Description:** This use case enables a user to reset their password if they have forgotten it
- **Actors:** User
- **Preconditions:** User has forgotten their password
- **Basic Flow:**
 - User clicks on the "Forgot Password" link on the login page
 - User enters their email address associated with the account
 - System sends a password reset link to the user's email
 - User follows the link and sets a new password
- **Postconditions:** User's password is successfully reset

Edit User Profile

- **Description:** This use case allows a user to modify their profile information such as name, email, etc.
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User navigates to the profile settings page
 - User makes desired changes to their profile information
 - User saves the changes
- **Postconditions:** User's profile information is updated in the system

Get Recent Activities

- **Description:** This use case enables a user to retrieve their recent activities tracked within the system
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User accesses the activities section of their account
 - System retrieves and displays the recent activities associated with the user's account
 - User receives a certain amount of XP
- **Postconditions:** User views their recent activities

Create a Team

- **Description:** This use case allows a user to create a new team within the system
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User navigates to the "Create Team" section
 - User provides necessary information for the team, such as name and description
 - User creates the team
- **Postconditions:** A new team is successfully created, and the user becomes the team captain.

Register to a Challenge

- **Description:** This use case enables a user to enroll themselves in a specific challenge within the system
- **Actors:** User
- **Preconditions:** User is logged in to their account.
- **Basic Flow:**
 - User browses available challenges
 - User selects a specific challenge to join
 - User registers for the selected challenge
- **Postconditions:** User is officially registered for the challenge. Upon completion of said challenge, user earns experience points and/or badges

Join a Team

- **Description:** This use case allows a user to join an existing team within the system
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User searches for available teams
 - User selects a team to join
 - User sends a join request to the team captain
- **Postconditions:** User becomes a member of the selected team upon approval from the team captain

View Leaderboard

- **Description:** This use case allows a user to view the leaderboards for challenges and activities within the system
- **Actors:** User

- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User navigates to the leaderboards section
 - User selects a specific challenge or activity type
 - System displays the relevant leaderboards
- **Postconditions:** User views the leaderboards for the selected criteria

Create Individual Challenges

- **Description:** This use case allows a user to create personalized challenges within the system
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User accesses the "Create Challenge" section
 - User defines the parameters and goals for the challenge
 - User creates the individual challenge
- **Postconditions:** User's custom challenge is added to the system

Create Streak

- **Description:** This use case allows a user to create a Streak for themselves
- **Actors:** User
- **Preconditions:** User is logged in to their account
- **Basic Flow:**
 - User accesses the "Create Streak" section
 - User defines the parameters for the Streak
 - User creates the Streak
- **Postconditions:** User's Streak is added to the system

Complete Streak

- **Description:** This use case allows a user to complete a Streak
- **Actors:** User
- **Preconditions:** User has fulfilled the conditions set for the Streak
- **Basic Flow:**
 - Streak is marked as completed
 - User is awarded experience points
- **Postconditions:** User is asked if they would like to repeat the Streak

Fail Streak

- **Description:** This use case allows a user to fail a Streak
- **Actors:** User
- **Preconditions:** User has not met the conditions set for the Streak
- **Basic Flow:**
 - Streak is marked as incomplete
- **Postconditions:** User is asked if they would like to restart the Streak

Get Experience Points

- **Description:** This use case allows a user to gain experience points every specified time interval
- **Actors:** User, Time
- **Preconditions:** The time interval has finished
- **Basic Flow:**
 - A snapshot is taken of the leaderboard for the finished time interval
 - Experience points are rewarded to each User depending on their leaderboard placement
- **Postconditions:** The leaderboard for the time interval is reset

Remove a User from the Team

- **Description:** This use case allows the team captain to remove a member from their team within the system
- **Actors:** Team Captain
- **Preconditions:** Team captain is logged in and has appropriate permissions
- **Basic Flow:**
 - Team captain accesses the team management section
 - Team captain selects the option to remove a team member
 - Team captain confirms the removal action
- **Postconditions:** The selected user is removed from the team

Register the Team to a Challenge

- **Description:** This use case allows the team captain to register their team for a specific challenge within the system
- **Actors:** Team Captain
- **Preconditions:** Team captain is logged in and has appropriate permissions.
- **Basic Flow:**
 - Team captain accesses the challenge registration section
 - Team captain selects the challenge for registration
 - Team captain registers the team for the selected challenge
- **Postconditions:** The team is officially registered for the challenge

Edit Team Profile

- **Description:** This use case allows the team captain to edit the profile information of their team within the system
- **Actors:** Team Captain
- **Preconditions:** Team captain is logged in and has appropriate permissions
- **Basic Flow:**
 - Team captain accesses the team profile settings.
 - Team captain makes desired changes to the team profile.
 - Team captain saves the updated team profile.
- **Postconditions:** The team profile is successfully updated in the system.

Create Global Challenges

- **Description:** This use case allows the company admin to create challenges that are visible across the entire system and automatically registers all users
- **Actors:** Company Admin
- **Preconditions:** Company admin is logged in and has appropriate permissions

- **Basic Flow:**
 - Company admin accesses the challenge creation section
 - Company admin defines the parameters and goals for the global challenge
 - Company admin creates the global challenge
- **Postconditions:** The global challenge is added to the system and is visible to all users

Manage Global Challenges

- **Description:** This use case enables the company admin to manage and oversee global challenges within the system
- **Actors:** Company Admin
- **Preconditions:** Company admin is logged in and has appropriate permissions
- **Basic Flow:**
 - Company admin accesses the global challenge management section
 - Company admin views existing global challenges and their details
 - Company admin can edit, delete, or update the status of global challenges as needed
- **Postconditions:** Global challenges are effectively managed and updated in the system

Manage Users

- **Description:** This use case allows the system admin to manage user accounts within the system
- **Actors:** System Admin
- **Preconditions:** System admin is logged in and has appropriate permissions
- **Basic Flow:**
 - System admin accesses the user management section
 - System admin views existing user accounts and their details
 - System admin can edit user information, deactivate accounts, or perform other administrative tasks
- **Postconditions:** User accounts are updated in the system.

Manage Levels

- **Description:** This use case enables the system admin to manage the user levels or tiers within the system
- **Actors:** System Admin
- **Preconditions:** System admin is logged in and has appropriate permissions
- **Basic Flow:**
 - System admin accesses the level management section
 - System admin views existing levels and their details
 - System admin can add, edit, or delete levels, as well as adjust the criteria for level progression
- **Postconditions:** Level structures are effectively managed and updated in the system

Manage Badges

- **Description:** This use case enables the system admin to manage badges awarded within the system
- **Actors:** System Admin
- **Preconditions:** System admin is logged in and has appropriate permissions
- **Basic Flow:**
 - System admin accesses the badge management section
 - System admin views existing badges and their associated criteria
 - System admin can add, edit, or delete badges, as well as adjust the criteria for badge acquisition

- **Postconditions:** Badges are updated in the system

Manage Activity Types

- **Description:** This use case enables the system admin to manage various types of activities tracked within the system
- **Actors:** System Admin
- **Preconditions:** System admin is logged in and has appropriate permissions
- **Basic Flow:**
 - System admin accesses the activity type management section
 - System admin views existing activity types and their details
 - System admin can add, edit, or delete activity types as needed
- **Postconditions:** Activity types are updated in the system