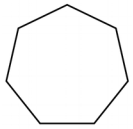
Computer Science & ENGINEERING

University of Nevada, Reno



Quantivity

Quantifying Productivity

Start developing discipline
by making better choices
using numbers.

Instructors:
Dr. Sergiu Dascalu
Devrin Lee

Team
Robin Brossard
Melina Tan

Tuesday, May 4th, 2021

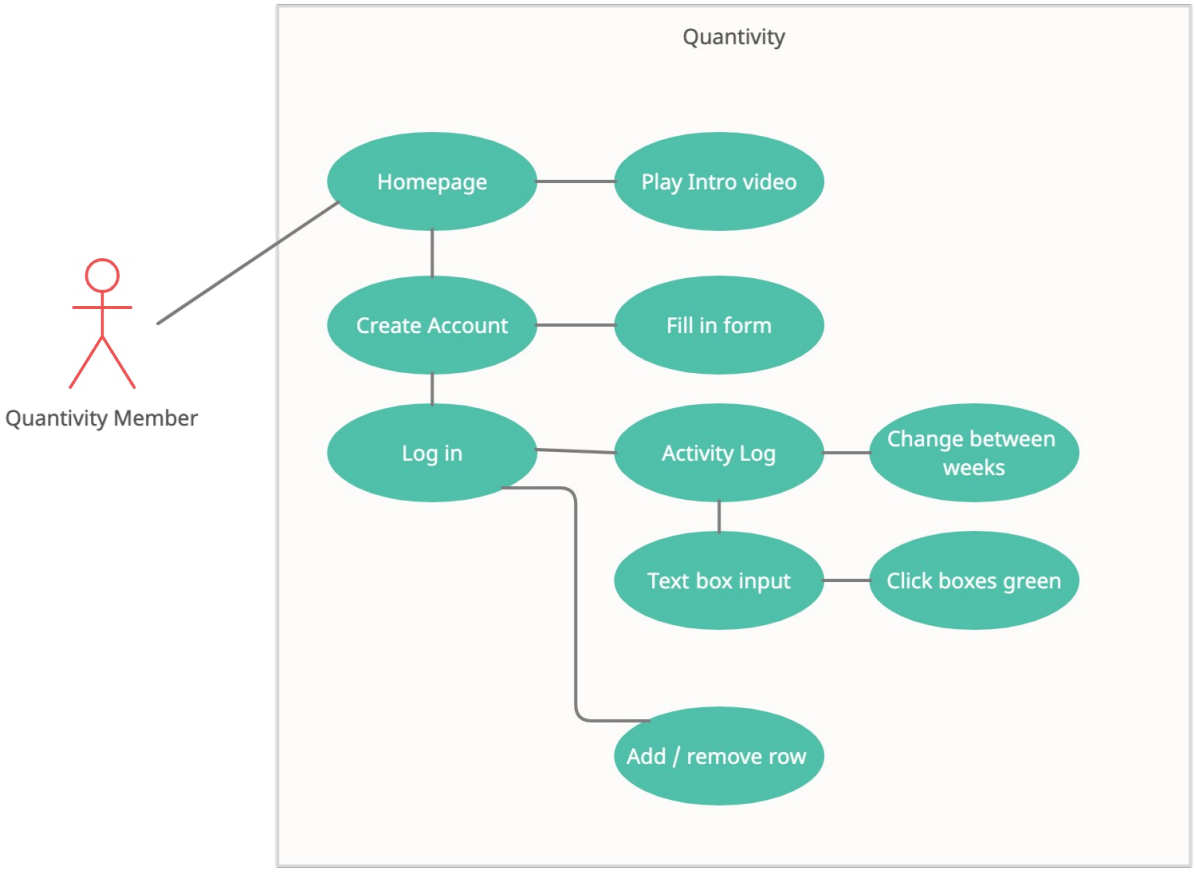
Table of Contents:

List of use cases and / or requirements 3

Project summary 4

Software overview 5

List of use case and / or requirements:



Project summary:

Project summary.

A. The use cases that were implemented were our homepage where it has an introduction video that can be played when clicked, a create account page, a login page for the user to log in with their credentials, then the activity login page which contains all the functionalities like adding / removing a row, editing a text box, and clicking the boxes green to indicate completion.

Fully developed requirements, and other capabilities:

Level 1	**FR1. The application will allow users to sign up and become a member in order to use the program.** **FR2. The application will allow a change of color of the day.** **FR3. The application will allow the ability to add colors (green and gray).** **FR4. The application will allow removal of certain colors for editing purposes.** **FR5. The application will allow input of certain colors / texts to define categories.** **FR6. The application will allow an update of the week to show current and past progress of the user.** **FR7. The application will allow us to save the data of the input from past days.** **FR8. The application will display a media clip entailing the purpose and instructions to use Quantivity.** **FR9. The application will allow import of project proposal pdf.** **FR10. The application will indicate spell check errors when typing certain texts.**

B. Functionality not implemented:

Level 2	**FR12. The application will allow a locking method on the user's current agenda to prevent any accidental changes.** **FR13. The application will allow an unlocking method to edit the user's current agenda.** **FR14. The application will have user specific password authentication for security purposes.** **FR15. The application will implement a user authentication to protect their own data.**
Level 3	**FR16. The application's user authentication will allow the user to only get a view of their own data.**

We did not have enough time to complete our level 2 and level 3 requirements due to only being two people working on this project for less than 4 months. A lot of the time went into planning and experimenting different frameworks and only about a month were spent towards the full implementation of level 1 to our project. FR10, spell checking, and FR12 and 13, locking and unlocking methods were the level 1 and 2 requirements we did not implement.

Contributions of team members:

Melina Tan: I was responsible for designing the prototype for Quantivity using InVision and crafting the introduction video on our webpage. I developed the UI of the homepage, UI of the activity log, as well as the UI for the Login page using ReactJS. Throughout the further updates of the UI as backend functionalities were being added, I had the help of Robin Brossard's assistance for making our website's UI more interactive. In the UI development process, I developed over 150+ lines of code on the first rough draft of the pages (Loginform.js, Navbar.js, Quansheet.js, the video upload on Homepage.js, as well as App.js.). Throughout the advancement process, Robin Brossard assisted in cleaning up the code and removing repeated messy code I've developed for a cleaner and functional UI that is compatible with backend functionalities.

Robin Brossard: I was responsible for designing the ExpressJS backend and integrating the backend with the ReactJS proxy. To do so I developed a restAPI to allow the React app

to query and update the database. In the beginning of the project, my primary focus was getting the frontend development off the ground and as the project progressed my focus shifted to the backend functionality. During the course of development I wrote 279,000 lines of code and deleted 136,000 lines of code. This number is abnormally high, because I forgot to include a gitignore file and accidentally uploaded quite a lot of library code to github.

Software overview:

Quantivity-RJS -> src:

App.css: All of the styling of our application and the activity log is in here.
App.js: All the routed files and libraries are included in here.
Homepage.js: The introduction video of quantivity is routed in here.
Index.css: The styling of the login page and create account page is in here.
(other files were included in the ReactJS package)

Quantivity-RJS -> src -> components:

Activitylog.js: It has the link to the login page as well as the Quansheet page.
CreateAccount.js: wrapper for the create account form and sends create account requests to the server
CreateAccountForm.js: Displays the create account form and receives user input
Login.js: It contains the backend functionalities of logging in process.
Loginform.js: It has development of the form layout on here.
NavBar.js: An upper left side navigation of the Homepage, login, and projectwebsite .
Projectwebsite.js: Did not have enough time to add projectwebsite.
Quansheet.js: The development of the UI of the activity and the functionalities that goes behind it are in here.

Server

app.js: router configuration and error handling

Server->bin

db.js: contains the code to initialize and configure the mongoose database.
userManagement.js: tracks which user is logged in and manages the auto-logout timer.
www: code entry point and general server configuration.

Server->routes

downloads.js: serves the design documents produced throughout the semester
index.js: default route
testAPI.js: simply notifies the user if the restAPI is functioning.
userData.js: handles get and post requests for userData, chiefly the main grid data
user.js: handles user creation and log in requests