National College of Ireland

Project Proposal

Stay Fit

12/10/2025

Final Year Project

Software Development

Academic Year: 2025/2026

Name: Alex Garbalyauskas

Student Number: 22440482

x22440482@student.ncirl.ie

Contents

1.0 Objectives

The **Stay-Fit Project** main goal is to develop a dynamic, user-friendly application accessible across multiple devices. Its primary purpose is to encourage daily fitness through an engaging and interactive user experience.

The application will allow users to create personal accounts and customize their fitness journey. Once the account is registered, users will receive **2–5 daily notifications** (based on their preference), each prompting them to complete a quick exercise for example, *"Do 10 push-ups", "Jump in Jacks, lets-go"* or *"try 30 squats, no break."*

To respond, users will record and upload **a short 30 to 60 second video** of themselves performing the exercise. After submitting on the application, they gain access to a real-time feed where they can view other users' submissions from around the world. This creates a fun, social, and motivating environment, as people get to see friends or even strangers completing the same daily fitness challenges in different places of the country, like schools, offices, or parks at the same time.

By combining fitness, daily videos, and social interaction, Stay-Fit transforms workouts into a shared experience, making exercises more enjoyable to complete. The goal is not only to promote consistent daily activity to stay fit but also to showcase they are not alone in their fitness goals and to make the application fun and humorous at the same time especially if the notification time is sent to a user while he's at work forcing him to record the video at his office grounds for everyone to see. The idea is to promote people to stay fit, active, healthy, and to share that to everyone either in a straight forward way but definitely in a comedic way as Charlie Chaplin said "A day without laughter is a day wasted".

2.0 Background

I decided to undertake the **Stay-Fit Project** because it addresses a personal long-time passion of mine: combining technology and fitness to create a more engaging and motivating workout experience that promotes being healthy and active where many are unmotivated to improve themselves. This project idea has been on my mind for some time, but I initially hesitated and delayed due to its technical complexity and the significant development time it would require to complete. However, with the final year project due this year, I believe the time is right to bring this idea to life hopefully for people to use if done correctly and on time.

To meet the objectives of section: 1.0, my plan is to focus on these functionality and areas.

1. **User Centric Design:** The app needs to be user friendly in its use and accessibility. I will try to make the design comfortable for the eye and the functions of use straight forward.

2. **Social interaction:** Since the application provides users notifications to post short-videos I think it will be essential to incorporate some sort of social feed/chat with emojis and reactions available for use to make the app enjoyable.

3. **Promoting Health:** The application will have maybe some web pages explaining how to perform certain exercises for users who are knew and maybe external links to other health, routines to follow. Likewise, the main subject theme of the project is to make more people engaged in fitness in a fun way.

4. **Humorous Content and Safety:** I will try to incentives doing the exercises in a comedic form of display, but not to the point of endangering themselves or anyone else, some safety notices and explanations on how to do the exercises will need to be included before they proceed with the exercises to post.

3.0 State of the Art

There are already many applications that promote well-being and fitness, each with its own unique niche and genre. The **Stay-Fit Project** shares similarities with these platforms but distinguishes itself through its focus on **real-time video challenges and community humour**.

One of the closest comparisons is the popular social app **BeReal**. In BeReal, users receive a daily notification prompting them to post a selfie, which is then shared simultaneously with friends or on a global feed. *Stay-Fit* follows a similar structure of time-sensitive notifications, but instead of selfies, users record **short videos of themselves completing a specific exercise such as a squat or push-ups**. The goal is to make fitness fun, comedic, social, and spontaneous, while keeping the same sense of shared timing and community that users can view either with their friends or with strangers.

Other similar applications that already exist include:

- **Motion** – A gamified fitness platform where users complete daily quests and weekly challenges, often alongside friends.
- **inKin** – A fitness challenge and corporate wellness app focused on teams, leaderboards, and group motivation.
- **Keep** – A fitness app that allows people to share workout and routine videos, with a large focus on exercise content and training.
- **My Fitness App** – One of the most popular fitness apps but if focuses more on calorie intake and meal planning than social inter-activeness and sharing.

Unlike these platforms, *Stay-Fit* blends the **spontaneity of BeReal** with the **fitness focus of Motion, inKin, and Keep**, while adding a distinct layer of humour, community**, and real-time participation**. The result is an app where users don't just track progress or share workouts—they laugh, stay consistent, motivated and experience fitness together.

4.0 Technical Approach

Development will begin on **October 1st** and will continue until **May**, with the outlined finalized project idea guiding the technical ideas of its functions. The application will be developed as a **web-based fitness application** with possible deployment for mobile devices depending on capability and time. My plan at the moment of writing this proposal is to work on the final year project every week, 2-3 days a week, since I have 1 days off one in the weekday and 2 on the weekends I will time manage as follows for the week.

1. **Monday:** College attendance, and continuous assessments.
2. **Tuesday:** College attendance, and continuous assessments.
3. **Wednesday:** Personal work and evening college work.
4. **Thursday:** Final year project development.
5. **Friday:** College attendance and maybe some project development.
6. **Saturday:** Final year project development.
7. **Sunday:** Final year project development.

This is not to be taken as definite as not everything can go as planned for every week, some days I might focus more on other college module work over the project and that's not including if I am tired or sick on some of these days or have unexpected change of plans.

Requirements

Requirements will be identified through:

- Brainstorming the ideas to technicality.
- Reviewing similar apps.
- Outlining the core functions (notifications, video upload, global social feed, reactions/emojis, user account management, sign-up and sign-out).
- Focusing on must have features, the essentials of my project over features that add on to the user experience.

Project Tasks to Be Completed

1. **Database:**
    - Create a database for user profiles, 30-60 second video submissions, notifications, comments and reactions.
    - Tech Considered: AWS, Firebase Storage, MongoDB.
    - Milestone: Functional schema implemented.
2. **Backend Development:**
    - Create user authentication, notification scheduling, video handling and postage.
    - Milestone: Working API endpoints for major features.
3. **Frontend Development:**

- Design a GUI/UI with responsive pages and routes to navigate the pages.
- Generate video uploads, social feed displays, end time reactions.
- Milestone: Interactive use cases completed.

4. **Landing/Information Pages**

    - Informative guide pages or weblinks explaining the application goals and intended usage.
    - Milestone: Informative page for users to read.

5. **Integration & Testing**

    - Combine backend and frontend together.
    - Functional, usability, and performance testing to make sure everything works.
    - Milestone: Stable beta version.

6. **Deployment (If Time Permits)**

    - Web app deployment on cloud hosting (e.g., Firebase, Heroku).
    - Mobile deployment.
    - Web store deployment.

7. **Documentation**

    - Technical documentation, user manual, and final project report.
    - Use cases, Diagrams, presentation and final video showcase.

5.0 Technical Details

The Stay Fit project will implement Typescript/JavaScript as its main language for frontend and backend, the principal libraries will be the React Native Expo, Firebase, expo android and other various unknown libraries, all of them will be downloaded from the terminal and added as an import to each of the component's files.

The **key algorithms or functions** decided and in consideration are as follows.

1. **Authentication/Profile:**
   Authentication for new user signups, saves email and password and adds a new profile. Automatic sign-ins if the profile is already set up.

2. **Sign Out:**
   Delete user profile when the account is no longer needed.

3. **Navigation:**
   Routes to different pages of the application such as home, friends, profile etc.

4. **Notifications:**
   A time clock that can be modified depending on how many the user wants. Sends 3-5 notifications per day to do an exercise. Each day of the week is a different set of exercises specialising towards a certain muscle group. E.g. on Monday user receives notifications to do chest exercises such as diamond push-ups, on Tuesday it will be legs, and on Wednesday back. This can be modified to the user's daily routine plan so if the user wants to do chest on Monday, they can set the chest exercises notification to go off on Monday and not on any other day.

5. **Video Upload:**
   After end-user receives the notification, they open up the phone camera place it safely on a surface for recording. Record themselves completing the exercise. The video can be recorded for 2 minutes max before the user can publish it onto the application. After publication they can view it themselves or their friends who befriended them on the app can watch the video. Users can modify the settings of the post for the wider public to see as well. Everyone who receives the post will post at the same time to make it fun, daily and spontaneous.

6. **Movement detector:**
   Camera detects movement to make sure the exercise is done correctly.

7. **Social Chat and Reactions:**
   Interact with users posts by adding live chat where users can add comments, emojis, likes and replies.

8. **Public Social Feed:**
   Users can post onto the public feed for everyone to see and like or comment on.

9. **Storage:**
   Save user posts on a storage database or cloud server.

10. **Undecided Features:**
    Light mode/dark mode for screens, profile manipulation, and feed settings manipulation.

6.0 Special Resources Required

To successfully complete this final year project these special requirements are of the most essential.

Hardware/Devices

- My personal laptop with optimum storage and power to create and store the project, helping in the creation, handling, testing and deployment of the project.
- Android mobile phone to handle, use and test the project.
- Database host to store my user inputted data for final usage and to test while in development.
- Cloud hosting for handling my project deployment, and to store data for usage and testing.
- Stable internet for testing, storage of data, and deployment.

Human Support

- I will need help from supervisor and mentors in the process of documentation and development to make sure everything is correct and aid in any spheres of work I might get stuck on.
- I will need other students or users to test out my application to make sure the authentication, storage, social feed, reactions, postages all work as intended.
- Helpful support to fix bugs, miscode or lack of knowledge to progress and finish the final year project faster.

Software/Tools

- Visual Studio Code, the main software development tool that I will use to create all of the code and components of my project.
- Database: Considered databases to store all the user data of their details and posts (Firebase), (MongoDB), (AWS), and (Azure).
- Backend frameworks such as (Node.js), (Next.js) to handle API's and notification handling.
- Frontend frameworks such as (React: Typescript, JavaScript), (React Native Expo, TypeScript, JavaScript), (HTML), (CSS) for web and mobile user UI functions that users will interact with.
- GitHub where all the code will be posted on for access and grading for everyone as an open-source project.
- Postman for API testing, Jest/React Testing Library or for frontend unit testing.

7.0 Project Plan

The project will follow a **phased plan** implementation from October to May, the focus will be to implement each function on a monthly basis and to test them out before continuing on with the next function goal.

Overall Plan:

Phase	Timeline	Key Deliverables
Research and Requirements	September-October	Main function and design thought out
Project Creation	October	Create the main folder of development
Initial Frontend	October-November	Main GUI/UI
Backend Development	October-May	Database structure, video upload API, and notification system
Frontend Development	October-May	Build responsive UI using React
Testing & Debugging	Continuous throughout	Testing each function and feature added
Documentation	Continuous throughout	Documenting each step, creating graphs, use-cases and writing down word documents

1. **Research and Requirements** (September-October)
    - Finalized beginners project plan, following the project plan step by step each week and month till completion and submission.

    - Redevleoping the plan as the production goes on.

2. **Main Project Draft** (October 1st)
    - Create the main folder of development of the project on Visual Studio Code, one folder for the backend and one folder for the frontend inside the folder “stay-fit”. Since it’s a beginner draft it might be scrapped to start a new one if better folder management is found or errors persists that might require a start over.

    - Create the frontend with application with React and backend with Node.js inside each respective folder.

3. **Initial Frontend Home Page** (October 1st-October 25th)

- Create the frontend application with React Native Expo with TypeScript as the main programming language as this application can be used on both web and android so it's the most suitable form of development to use for my project idea.

- Install all the libraries that is needed for the frontend while in development these will be discovered during the development.

- Create the index/home.tsx file where the main frontend development will begin.

- Design the initial frontend with a friendly UI design.

4. **Authentication** (October 1st-October 18th)
    - Create a Firebase project, a software cloud tool found accessible online where users can create authentication sign-ups and database storage.

    - Create the sign-in and sign-up.tsx files with the added firebase console authentication code imports and logic for the authentication function.

    - Route the sign-up and sign-in pages to the home page after the successful login is complete.

    - Save user details on Firebase and make sure the sign-in page saves the user data so if the users leave's the homepage their email is automatically saved on the sign-in page making the sign-in quicker next time.

5. **Navigation Routes/Navbar** (October 1st- October 26th)
    - Create a navbar typescript file component that routes each file page to make sure everything is interconnected

    - Design the navbar to look to have nice navigation options on the lower side of the device screen where you can open the home, friends, post, chat and profile page.

6. **Post page and Functionality** (November 1st – November 20th)
    - Create the post page.tsx file with the camera recording functionality that lets users to record themselves and to post it after.

    - Create the posting settings page where users put in their details on how long the video will be, the title, option to share with friends only or for the wider public and comments before posting the video.

7. **Movement Detector** (November 1st – December)
    - Implement a detector on the camera that verifies that movement is occurring so that the exercise is validated before submission.

8. **Friends Page** (November 1st-December)
    - Design the frontend friend expo route page where users can search for other authenticated members on the app to befriend and share with.

9. **Connect with Friends** (November 1st – December)
    - Add the required logic that allows users to search and connect their profiles with other user profiles.

10. **Social Chat/Feed** (December – January)
    - Creation of the frontend chat file that allows users to send messages to friends and vice versa.

    - Connect the backend of the chat to the frontend by creating a chat server.

    - Likewise, create the chat as a separate project, deploy it onto GitHub and connect it with an I-frame to the main project.

11. **Profile Settings** (January – February)
    - Addition of the profile settings page on the frontend with options to logout, delete account, manipulate profile picture and more.

12. **Database Storage** (October – May)
    - Verifying that the newly created video files are saved on Firebase, if any issues will occur with storage a new form of storage will be needed to replace the intended one.

    - Continuous testing to keep functionality intact.

13. **Notification System** (December – March)
    - Device notification system set in place that can be modified as the user intends that sends 3-5 notifications per day, implement into the backend and frontend for the application.

14. **Documentation and Monthly Reports** (Continuous)
    - Write down the process of development, tools used, and new plans proposed or implemented.

    - Finish each month's report outlining what was done each month of the development cycle.

- Design, draw graphs and charts for the documentation.

15. **GitHub** (Continuous)
    - Push and submit the developed code onto GitHub for code review and validation.

16. **Final Testing and Polishing** (May)
    - Validating that everything works, all routes connected, backend and frontend connected, all functions work smoothly as intended. Testing device workability, testing with multiple users, testing storage database. Finishing various other minor tests before final submission.

17. **Deployment** (May)
    - Deploy the final project onto GitHub link or Vercel to have a fully deployed project link to share with.
    - Maybe deployment onto the app store or other android deployments for download.

8.0 Testing

The system will be evaluated through a testing process incorporating both technical and user-focused methods. The goal is to ensure the system performs reliably in all instances, securely, and meets user usability expectations without complications on how to use the application.

Testing Approach

1. **Unit Testing:**
    - Each component file (e.g., login form, upload post) will be individually tested.
2. **Integration Testing:**
    - Ensure backend and frontend connected correctly.
3. **Device Testing:**
    - Test across devices (laptop, mobile browser, Android app).
4. **Performance Testing:**
    - Evaluate video upload speed, feed load times, and server response.
5. **Security Testing:**
    - Verify encryption of user credentials, encryption for user chat, secure file storage and secure uploads.
6. **Usability Testing:**
    - User testing sessions with peers to gather feedback on interface clarity, ease of navigation, and engagement.
    - Adjust design based on participant input while maintaining ethical data use (no personal data will be stored or shared beyond testing purposes).

Ethical Considerations

- All participants involved in testing will be informed and provide consent.
- No sensitive data will be stored during testing.
- No sensitive information and data will be shared without consent.
- Video submissions for testing will be limited to sample files which will be deleted.