*An exercise app that elicits a sense of community for users who are new and used to going to the gym.*

*Team Tigahs*

Project Portfolio

*May 3rd, 202*

Introduction 2

Team Tigahs 3

System Requirements 3

User Stories 3

User Story #1 3

User Story #2 3

Project Management 3

Continuity of Operations Plan (COOP) 3

Project Plan

System Architecture Design and Development

System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>

Project Postmortem <Postmortem>

Project Wins

Root Cause Analysis

Lessons Learned

System Design <Milestone 2: System Architecture> 1

System Architecture <Milestone 2: System Architecture> 1

Component Design 1

Data Flow 1

System Components <Milestone 3: System Implementation> 1

Component [Component Name 1] 1

Component [Component Name 2] 1

Component [Component Name n] 1

Design Pattern <Milestone 3: System Implementation> 1

Design Pattern <Milestone 3: System Implementation> 2

System Implementation <Milestone 3: System Implementation> 3

[Project portfolio template directives and placeholders (delineated by “[ ]” or “< >” and/or highlighted or optional sections not included) should be removed from the document prior to submission. Empty sections for inclusion in later submissions may remain in the document for early submissions.]

[IMPORTANT: All diagrams developed using Enterprise Architectures must include the following acknowledgement: “Thanks to SPARX Systems for LSU student and faculty use of Enterprise Architect for academic purposes”.]

# Introduction

One of the biggest challenges people face when trying to start working out is figuring out what exactly they need to do to start out on the right foot. Attempting to learn it all on your own can be very intimidating, and it has led many people to give up on even trying to work out at all. New gym-goers aren’t the only group of people who struggle with going to the gym either. Experienced people also struggle to get to go to the gym consistently, to work out properly, and to find a sense of community that helps them to feel like they belong.

Our goal for this app is to create a social fitness app that matches users based on their fitness goals, workout routines and schedules. The app would allow users to search and connect with workout partners, trainers and like-minded individuals in their local gym or fitness community. In order to make this happen, we may make use of technologies such as as JavaScript, Flutter, Dart, CSS, or HTML to create a website and a mobile apps while also using MySQL for database management. Java or C# will more likely than not be our OOP language of our choice for connecting the website and mobile apps to the database for retrieval and input of user data.

Core Features:

* Profile Creation: Users will be able to create a profile that includes general information such as their age, sex, height, weight, etc. A profile picture and a bio will also be provided, should the users want to include more information. On top of height and weight, users can also supply a series of basic exercises so others have an idea of what their physical build is by analysing their individual muscle groups and body as a whole.
* Matching Algorithm: The tool would use a matching algorithm to connect users with workout partners who share similar stats and location. The algorithm will be customizable so that they can choose whether they want to meet people who are more or less similar in physical stats.
* Messaging: Based off the user matches derived from the algorithm, users will be able to chat with others to allow them to meet up with each other.

Viable Features:

* Social Feed: The app could have a social feed where users can post about their workouts, share progress, and connect with other users. The feed could also be used by trainers to share tips and insights on fitness and wellness.
* Workout and Training Plans: The app could provide users with personalized workout plans and training programs tailored to their fitness goals and level of experience. Trainers could also create and share custom training plans with their clients.

Stretch Features

* In-app Purchases: The app could offer premium features, such as customized workout and training plans, access to exclusive workout classes, and personalized coaching from trainers, for a fee.
* The app could also have a calendar feature to help users keep track of their scheduled workouts.
* Safety Features: The app would need to have safety features including but not limited to guidelines for meeting up with workout partners, and reporting tools to report any suspicious behavior.

# Team Tigahs

Aidan Eiler – Team Leader, Project Owner, System Architect, Backend Developer

Jonathan Morse – SCRUM Master, Frontend Lead

Tiffany Scroggins – Frontend Developer

Tyler Stephens - Frontend Developer

Reece Hernandez  - Frontend Developer

Jack Legnon – Backend Developer

Project GitHub Link: https://github.com/Gr8Potato/CSC-3380-Project

# System Requirements

## User Stories

### User Story #1

As an average gym goer, I want to be able to connect with like-minded individuals of similar physique (stats, build, etc.), so I can get the best results and socialize with other people.

### User Story #2

As a person interested in working out, I want to be able to network with trainers and other gym enthusiasts, so I can learn from someone with experience.

# Project Management

## Continuity of Operations Plan (COOP)

We have created a Discord server which we will use for accessible communication, both text, audio, and video. We also will have weekly or bi-weekly in-person meetings to update one another on progress as well as to discuss any issues we may be having. We're considering practicing paired programming, in order to ensure everyone has the capability to produce quality work, as well as dividing work amongst the team, attempting to cater towards our natural skill sets. Given our team has an odd number of members, it may be the case that a group of three is formed or one person works on a particular component individually.

Using Discord as our primary form of communication throughout the project allows us to be very flexible should unforeseen issues arise. If we can’t meet in person for an extended period of time, we will use our Discord's audio chat feature in place of our in-person meetings. If, for whatever reason, it's the case that one or more of our members is unable to contribute to the project, whether it be temporarily or permanently, we can expect the other in the programming pair to take over the work of whoever can't, provided the work doesn't become arduous, and if that become the case, we will begin adapting to fulfill the project's goals and its trajectory.

## Project Plan

### System Architecture Design and Development

[Milestone 1 (Proposal): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 2 (Architecture). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 2 (Architecture): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estima-ted**  **Effort (hrs)** | **Actual**  **Effort** | **Estimated**  **Start Date**  **(MM/DD/YYYY)** | **Estimated**  **Finish Date**  **(MM/DD/YYYY)** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| 1.0 | Learn EA. |  |  |  |  |  |  |  |
| 1.1 | Attend Class on Thursday. |  | 1.5 | 1.5 | 2/9/2023 | 2/9/2023 | 2/9/2023 | 2/9/2023 |
| 1.2 | Have all members complete EA assignment(s). |  | 5 | 3 | 2/9/2023 | 2/10/2023 | 2/10/2023 | 2/12/2023 |
| 1.3 | Spend time practicing EA outside of homework hours. |  | 5 |  | 2/10/2023 | 2/11/2023 | 2/10/2023 | 2/11/2023 |
| 2.0 | Determine System Architecture |  |  |  |  |  |  |  |
| 2.1 | Determine which components might be needed |  | 10 | 1.5 | 2/10/2023 | 2/13/2023 | 2/10/2023 | 2/13/2023 |
| 2.2 | Evaluate the benefits and drawbacks of each system architecture | 1.1 | 10 | 0.5 | 2/13/2023 | 2/14/2023 | 2/13/2023 | 2/13/2023 |
| 3.0 | Integrate System Architecture to Project Design |  |  |  |  |  |  |  |
| 3.1 | Determine data flow | 2.1,2.2 | 10 | 2 | 2/14/2023 | 2/17/2023 | 2/13/2023 | 2/16/2023 |
| 3.2 | Generate a rough draft for different machine states | 2.1,2.2 | 15 | 2 | 2/17/2023 | 2/20/2023 | 2/13/2023 | 2/16/2023 |
| 3.3 | Refine components if need be | 3.2 | 3 |  | 2/20/2023 | 2/21/2023 | 2/17/2023 |  |
| 3.4 | Integrate Fully Fleshed System Architecture into EA | 3.3 | 25 |  | 2/21/2023 | 2/23/2023 |  |  |
| 3.5 | Begin prototyping | 3.4 | 8 |  | 2/22/2023 | 2/23/2023 |  |  |

### System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>

[Milestone 2 (Architecture): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 3 (System Implementation). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 3 (System Implementation): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
|  |  |  |  |  |  |  |  |  |

## Project Postmortem <Postmortem>

### Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

### Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions. ]

### Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

# System Design <Milestone 2: System Architecture>

[*Include a short (1-2 sentences) statement about the system design*.]

## System Architecture <Milestone 2: System Architecture>

[*A short description of the system architecture.*]

### Component Design

[*Insert image of system architecture component diagram. Include the name of the team member that created the diagram in EA.*]

[*Architecture overview, to include user I/O, external data sources, and major system components.* ]

### Data Flow

[*Insert image of system architecture data flow diagram. Include the name of the team member that created the diagram in EA.*]

[*Architecture data flow discussion: a high-level description of the data between both internal major components and external data sources.*]

## System Components <Milestone 3: System Implementation>

[*Include a component sub-section for each component in the architecture diagram. Each component subsection will include a class diagram*]

### Component [Component Name 1]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

### Component [Component Name 2]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

### Component [Component Name n]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least design patterns covered in class. Include the name of the team member that created the diagram in EA.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. Include the name of the team member that created the diagram in EA. A second design pattern may be included for bonus points.*]

# System Implementation <Milestone 3: System Implementation>

[*In the table below, include a row for each component in your System Architecture diagram. In the second column, list the programming language(s) used to implement the component and the what % of that programming language is used in the implementation. In the third column, list the team member(s) that implement the component and what % of that implementation was completed by that team member. IMPORTANT NOTE: All architectural components must be implemented by an object-oriented programming language: Java, C++, or C#.*]

|  |  |  |
| --- | --- | --- |
| **Architectural Component** | **Programming Language(s) %** | **Team Member(s) %** |
| *[Data Manager]* | *[C++ (45%)*  *Java (55%)]* | *[Mickey Mouse (15%)*  *Donald Duck (20%)*  *Daisy Duck (40%*  *Pluto (25%)]* |