**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

Source for this template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

# 1.0 Introduction

The Introduction section provides an overview of the system using software requirements analysis and design for the scope of the system.

## Purpose

This Requirements Analysis Document serves as a comprehensive specification for GymBuddies, a mobile app dedicated to connecting fitness enthusiasts and facilitating workout partnerships in Toronto. This document provides the development team with detailed requirements to ensure each phase of design, development, and testing aligns with user expectations and business objectives. It will act as a baseline reference for the project, detailing what the app must achieve without specifying the implementation details.

## Scope

The scope of the GymBuddies application focuses on enabling users to connect with compatible workout partners based on fitness goals, schedules, and location. It includes features that enhance social engagement within a fitness-focused community. The app will cater to both iOS and Android platforms and will provide a range of functionalities from profile creation to geolocation-based matching.

* **In Scope:**
  1. **User Registration and Profile Creation:** Users can register using email or phone numbers and create detailed profiles that include fitness goals, experience level, and workout preferences.
  2. **Geolocation Services:** Integrated GPS functionality enables users to find nearby workout partners and explore nearby gyms, leveraging Google Maps for accurate location-based recommendations.
  3. **Filter-Based and Swipe-Based Matching System:** Users can apply filters to find partners by fitness level, preferred activities, and proximity. The app also supports a swipe-based matching interface to make partner selection intuitive and engaging.
  4. **In-App Messaging and Communication:** Matched users can communicate through secure in-app messaging to coordinate workout plans and stay connected.
  5. **Workout Tracking and Challenges:** The app allows users to log and track workouts, set personal fitness goals, and participate in challenges with friends for additional motivation.
  6. **User Verification and Reporting System:** To enhance user safety, the app includes a verification system that uses ID-based selfies and allows users to report suspicious or inappropriate behavior.
  7. **Integration with Google Maps:** Users can explore gym locations, view gym facilities, and read reviews, all through Google Maps integration.
  8. **Backend Infrastructure and Database Management:** The app will utilize Firebase for backend support, providing real-time data storage and synchronization capabilities to handle user data and interactions effectively.
  9. **iOS and Android Mobile Application Development:** The project includes developing native mobile apps for both iOS and Android, focusing on a seamless user experience and adherence to app store guidelines.
* **Out of Scope:**
  + **Integration with Fitness Tracking Devices or Third-Party Apps:** The initial version of GymBuddies will not include data synchronization with external fitness devices or applications.
  + **Implementation of a Payment System for Premium Features:** Although future monetization options may be explored, there is no integration with payment systems or premium features in this release.
  + **Development of a Web-Based Version:** The app will only be available as a mobile application, with no plans for a web version in this phase.
  + **Social Media Integration:** There will be no direct integration with social media platforms.
  + **Creation of Content (Workout Plans or Nutritional Advice):** The app will not generate or offer built-in fitness plans, workout routines, or dietary advice.

# System Overview

GymBuddies is a mobile application designed to connect individuals who share fitness goals by helping them find compatible workout partners based on interests, schedules, and location. The application aims to foster a supportive fitness community, enhancing motivation and accountability among users. GymBuddies leverages geolocation, profile matching, and an in-app messaging system to provide a streamlined platform where users can discover workout partners, track their fitness progress, and participate in community challenges.

## Project Perspective

GymBuddies is a standalone, self-contained application designed to address the gap in social fitness support within urban areas, particularly Toronto. Many existing fitness apps focus on personal tracking or generalized social features; however, GymBuddies is specifically tailored to match users with nearby workout partners, helping them build local fitness networks. GymBuddies combines elements of fitness tracking with social connectivity, offering features like geolocation-based partner matching and workout logging to support a fitness community focused on mutual motivation.

GymBuddies replaces traditional ways of finding workout partners, which often rely on social circles or online groups, by creating a more accessible, user-friendly solution that directly connects fitness enthusiasts. It seeks to create a digital space where fitness goals can be shared, tracked, and celebrated collectively, making fitness a more engaging and collaborative experience.

## System Context

GymBuddies integrates into the broader fitness and wellness landscape by offering a platform that bridges the gap between solo workouts and community-based activities. The application meets the modern demand for both digital fitness support and social interaction, providing a safe and accessible space for users to find workout partners, track progress, and challenge themselves. By offering features like profile-based matching, geolocation services, and secure communication channels, GymBuddies enables users to build meaningful fitness relationships. The app promotes a healthier, community-oriented lifestyle in urban environments and encourages consistency and engagement in personal fitness routines through social accountability.

## General Constraints

The following constraints will impact the design, development, and implementation of the GymBuddies application:

* **Platform Compatibility**:
  + GymBuddies must be compatible with both iOS and Android platforms, requiring cross-platform development efforts. This constraint may introduce complexity in ensuring a consistent user experience across different devices and operating systems.
* **Technology Stack**:
  + The chosen technology stack, primarily Firebase for backend infrastructure and Google Maps for location services, must support real-time updates and scalability. This selection may limit the flexibility of using alternative services and requires thorough performance testing to ensure the stack can handle high user loads without compromising responsiveness.
* **Data Privacy and Compliance**:
  + The application must comply with PIPEDA (Personal Information Protection and Electronic Documents Act) regulations, which govern data privacy standards in Canada. Compliance will impact how user data, including geolocation, is stored, processed, and shared, requiring secure data encryption and privacy-preserving data handling practices.
* **Budget Constraints**:
  + Development, testing, and marketing efforts must adhere to a fixed budget. Financial limitations may restrict the scope of features, marketing outreach, or the ability to hire specialized talent, potentially affecting both the quality and timeline of project deliverables.
* **Time Constraints**:
  + The GymBuddies app must be completed by March 28, 2025, necessitating adherence to project timelines and milestones. Any delays in meeting critical deadlines could impact the product launch and early user adoption, particularly if missed during peak market periods.
* **Integration with Third-Party APIs**:
  + GymBuddies relies on third-party APIs (e.g., Google Maps) for essential functionalities like geolocation. API limitations, such as rate limits and service availability, may affect app performance, requiring robust error handling and fallback mechanisms to ensure functionality.
* **Scalability for Future Expansion**:
  + The application must be designed with scalability in mind to accommodate a growing user base and potentially expanded features. Meeting this requirement places constraints on architectural design, necessitating a modular approach that can support future growth without significant rework.

## Assumptions and Dependencies

**Assumptions**:

1. **Resource Availability**:
   * The necessary resources for hosting and maintaining the application (e.g., Firebase for backend support) will be available and sufficient for the expected user load.
2. **Technology and Development Tools**:
   * Development tools, including Android Studio with the latest SDK for Android and Swift for iOS, will remain accessible and fully supported throughout the project.
3. **Feature Implementation Feasibility**:
   * It is assumed that all planned features, including profile management, geolocation, and in-app messaging, are technically feasible and can be implemented within the project’s timeline and budget.
4. **Google Maps API for Location Services**:
   * Google Maps will be available and sufficient to support location-based functionalities like partner and gym identification.
5. **Privacy Regulation Compliance**:
   * The app will comply with PIPEDA, and any necessary adjustments to meet these standards will be achievable without major redesigns.
6. **App Store Compliance**:
   * The app will adhere to the guidelines of the Google Play Store and Apple App Store, and will not encounter major obstacles during submission and review processes.

**Dependencies**:

1. **Third-Party Services (Google Maps, Firebase)**:
   * GymBuddies relies on external services for core functionalities (e.g., Google Maps for location and Firebase for real-time backend support). Any changes, outages, or limitations in these services could impact app performance or functionality.
2. **App Store Guidelines**:
   * Compliance with app store guidelines is essential for app distribution. Changes in policies or additional requirements from the app stores could introduce unexpected delays or necessitate design adjustments.
3. **User Internet Access**:
   * The app requires users to have internet access for real-time updates, location-based features, and in-app messaging. Limited or unstable network access among users may affect their experience, particularly in areas with poor connectivity.
4. **Compliance with Privacy and Security Regulations**:
   * As user data security is paramount, GymBuddies must adhere to PIPEDA, which influences how data is collected, stored, and managed. Any regulatory changes could impact the app’s compliance and require modifications to data management protocols.

## 3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.1 <Functional Requirement or Feature #1>

1. **User Registration and Profile Management**

* **Description**: GymBuddies users must be able to easily create and manage profiles that reflect their fitness preferences and goals.
  + **Inputs**: User details such as name, age, fitness level, goals, and location.
  + **Processes**: Collect user data and validate entries, including optional photo ID for verification.
  + **Outputs**: A completed user profile displayed in the system, available for matching and interaction.
  + **Notes**: Profiles should be editable, and optional verification should enhance user trust

1. **Geolocation-Based Partner Matching**

* **Description: GymBuddies will match users based on their location, allowing them to find nearby workout partners with similar fitness levels and goals.**
  + **Inputs**: User’s GPS location, fitness level, and preferences (e.g., workout type, availability).
  + **Processes**: Locate users within a specified radius, apply filters based on matching preferences, and rank partners based on proximity.
  + **Outputs**: A list of nearby users who match the specified criteria, presented in the app for easy browsing.
  + **Notes**: Location updates in real-time to maintain accuracy; integrates with Google Maps for displaying gym locations.

1. **Filter-Based and Swipe-Based Matching System**

* **Description: Users can apply custom filters or use a swipe feature to browse through potential workout partners, enhancing flexibility in partner selection.**
  + **Inputs:** Filters such as age range, fitness goals, preferred workout times, and specific activities.
  + **Processes:** Filter and sort the partner list based on user preferences; swipe feature for quick selections.
  + **Outputs:** Filtered list of potential workout partners with options for connection requests.
  + **Notes:** Filters support both casual and advanced workout preferences, catering to a broad user base.

1. **In-App Messaging System**

* **Description: GymBuddies offers a secure messaging platform for matched users to communicate and plan workouts.**
  + **Inputs**: Messages sent by matched users.
  + **Processes:** Encrypt messages, store in the database, and deliver in real-time.
  + **Outputs:** Updated chat interface with new messages visible, including notifications for new chats.
  + **Notes:** Push notifications are provided for new messages to enhance user responsiveness.

1. **Workout Tracking and Progress Sharing**

* **Description: Users can log their workouts and view the workout activity of their buddies, encouraging consistent fitness routines.**
  + **Inputs:** Workout data such as workout type, duration, and location, manually entered by the user.
  + **Processes:** Store and update workout logs, generate progress summaries, and display buddy activity.
  + **Outputs:** A visual workout log for each user, accessible on their profile and shared with workout buddies.
  + **Notes:** This feature aims to motivate users by displaying their own and their partners' fitness activities.

1. **User Verification and Reporting System**

* **Description: A verification and reporting system to ensure user safety by allowing ID verification and enabling users to report or block inappropriate profiles.**
  + **Inputs:** Verification photos, report submissions by users.
  + **Processes:** Verify user photos for identity confirmation, flag reported users for manual review.
  + **Outputs:** Verified user profiles marked as trustworthy; flagged profiles held for administrative review.
  + **Notes:** This system is essential to maintaining a safe, supportive community within GymBuddies.

1. **Google Maps Integration for Gym Locations**

* **Description**: **The app integrates with Google Maps to display nearby gyms and fitness facilities for users seeking convenient locations.**
  + **Inputs**: User’s location data.
  + **Processes**: Fetch nearby gym locations from Google Maps, display options within a set radius.
  + **Outputs**: A list of nearby gyms visible within the app, mapped for easy navigation.
  + **Notes**: Users can click on gym locations for more details, including amenities and reviews, supporting informed decisions.

## 3.2 Use Cases

### 3.2.1 Use Case #1 ... User Registration and Profile Setup

* **Actors**: New User
* **Preconditions**: The user has downloaded GymBuddies and opened the app for the first time.
* **Postconditions**: A completed user profile is created and saved in the system.

**Main Flow**:

1. The user opens the app and selects "Sign Up."
2. The user enters required details (name, age, email, fitness level, goals).
3. The app prompts the user to upload a profile photo (optional for verification).
4. The user submits the information, and the system validates all entries.
5. If all information is valid, the system creates the user profile and confirms account creation.

### 3.2.2 Use Case #2 ... Finding and Matching with Workout Partners

* **Actors**: Registered User
* **Preconditions**: The user has an active profile with location services enabled.
* **Postconditions**: A list of compatible workout partners is presented to the user.

**Main Flow**:

1. The user selects the "Find Workout Partners" option.
2. The app requests location access (if not already enabled) and retrieves the user’s location.
3. The user applies filters (e.g., proximity, fitness level, workout preferences).
4. The system searches for nearby users who match the criteria.
5. The app displays a list of compatible workout partners, with options to send a connection request.

### 3.2.3 Use Case #2 ... Messaging a Workout Partner

* **Actors**: Matched User
* **Preconditions**: The user has matched with a workout partner.
* **Postconditions**: A message is sent and received by the matched partner.

**Main Flow**:

1. The user opens the "Messages" section.
2. The user selects a matched partner from the list of conversations.
3. The user types a message and presses "Send."
4. The system securely stores and sends the message.
5. The partner receives a notification of the new message, which appears in their chat interface.

### 3.2.4 Use Case #2 ... Tracking and Logging a Workout\

* **Actors**: Registered User
* **Preconditions**: The user is logged into their account.
* **Postconditions**: The workout is saved to the user's activity log and visible on their profile.

**Main Flow**:

1. The user selects "Log Workout" from the main menu.
2. The user enters workout details (type, duration, location).
3. The system validates the data and confirms entry.
4. The workout is saved to the user’s activity log and shared with their buddies.
5. The app updates the user’s profile with the latest workout information.

### 3.2.5 Use Case #2 ... Reporting a Suspicious Profile

* **Actors**: Registered User
* **Preconditions**: The user encounters a profile that they want to report.
* **Postconditions**: The reported profile is flagged for review by the admin team.

**Main Flow**:

1. The user navigates to the profile they wish to report.
2. The user selects the "Report Profile" option.
3. The app prompts the user to select a reason for reporting.
4. The user submits the report.
5. The system flags the profile, records the report, and notifies the admin team for further action.

## 

## Data Flow Diagram

**3.3 Data Modelling and Analysis**

* Normalized Data Model Diagram
* A screenshot of a computer

  Description automatically generated
* Activity Diagrams
* A diagram of a company

  Description automatically generated
* Sequence Diagrams
* A screenshot of a computer

  Description automatically generated
* UML Class Diagram
* A screenshot of a computer

  Description automatically generated

**3.4 Process Modelling**

## 4.0 Non-Functional Requirements

**4.1 Performance**

* **Response Time**: The system should respond to 95% of user actions, such as opening a profile or loading messages, within 2 seconds.
* **Load Handling**: GymBuddies should support up to 10,000 concurrent users without significant degradation in performance.
* **Data Retrieval**: Loading partner profiles, nearby gym locations, and workout logs should be completed within 3 seconds for 90% of queries.
* **Messaging Delivery**: In-app messages should be sent and received in under 1 second to ensure smooth, real-time communication.

**4.2 Reliability**

* **System Uptime**: The application should maintain a minimum uptime of 99%, ensuring continuous availability for users.
* **Data Integrity**: User data (e.g., workout logs, profile information, message history) must be accurately saved and retrievable within 5 minutes of data entry.
* **Error Logging**: All system errors must be logged automatically, with critical errors reported to administrators within 10 seconds for prompt resolution.
* **Recovery from Failures**: In the event of a failure, the application should recover and restore normal operations within 1 hour, with minimal impact on users.

**4.3 Availability**

* **24/7 Access**: The application should be available at all times, providing users with uninterrupted access to their profiles, matches, and messages.
* **Maintenance Notifications**: Users must be notified of planned maintenance at least 24 hours in advance, minimizing the impact on their experience.
* **Redundancy**: The application infrastructure should incorporate redundancy to avoid single points of failure, using cloud-hosted backup services to ensure continuity.

**4.4 Security**

* **Data Encryption**: All sensitive user data, including location and messages, must be encrypted both in transit and at rest to protect against unauthorized access.
* **Authentication**: GymBuddies should provide two-factor authentication as an optional security feature for enhanced user account protection.
* **Compliance**: The application must comply with PIPEDA regulations to ensure secure handling of user data, adhering to Canadian data privacy laws.
* **User Verification**: Optional photo ID verification should be implemented for users to enhance community trust and safety.
* **User Reporting System**: The system must allow users to report and block other users, with reports managed by an admin panel for timely review and action.

**4.5 Maintainability**

* **Code Documentation**: All code must be thoroughly documented to facilitate troubleshooting and updates by the development team.
* **Modular Design**: The system should be designed modularly, allowing updates and feature expansions with minimal impact on other components.
* **Bug Resolution Time**: Critical bugs must be resolved within 4 hours of identification, while minor bugs should be fixed within 48 hours to ensure stability.
* **Scalability**: The architecture should support scalability to accommodate a growing user base and expanded features, such as additional fitness tracking tools.

**4.6 Portability**

* **Cross-Platform Compatibility**: GymBuddies should deliver a consistent user experience across iOS and Android platforms.
* **Data Transfer**: The system must support secure and seamless data transfer for backups and potential migration to new cloud services, ensuring continuity of service.
* **Responsive UI**: The application interface should be responsive and compatible with a range of screen sizes and resolutions, optimizing usability across different devices.

## 5.0 Logical Database Requirements

## Firebase provides real-time data synchronization, making it perfect for features like chat, live GPS tracking, and instant updates on user activities. It supports seamless integration across Android, iOS, and web, ensuring consistent functionality for users on different platforms. Moreover, Firebase includes secure, pre-built authentication methods like email/password, social logins, and anonymous login, simplifying user management.

## 6.0 Other Requirements

Additional requirements, if any.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Team Member, Production | Nhu Nam Nguyen | Nhu Nam | 11/15/2024 |
| Team Member, Production | Duc Thien Tran | Duc Thien | 11/15/2024 |
| Team Member, Production | Ronan Behan | Ronan | 11/15/2024 |
| Team Member, Production | Simon Vu | Simon | 11/15/2024 |