System Specification - PhotoMap

By Paulson Kimani

CSC 3150: Systems Design

Andy Cameron

Spring 2024

June 2, 2024

Table of Contents

1.0 Executive Summary 4

2.0 Introduction 5

2.1 Problem Statement / Project Vision 8

2.2 System Capabilities 10

2.3 Non-functional Requirements and Design Constraints 11

2.4 System Evolution 12

2.4.1 Version 2 Changes 13

2.4.2 Version 3 and Beyond Changes 19

2.5 Document Online 27

3.0 Structural Model 29

3.1 Model Introduction 29

3.2 Class Diagrams 30

3.3 Metadata 31

3.3.1 Location 32

3.3.2 Map 34

3.3.3 Photo-Shoot 36

3.3.4 Profile 38

3.3.5 User 40

4.0 Architecture Design 42

4.1 Architecture Overview 42

4.2 Infrastructure Model 44

4.2.1 Deployment Diagram 1 – Architecture Overview 44

4.2.2 Deployment Diagram 2 – Nodes and Artifacts 45

4.3 Hardware and Software Requirements 47

4.3.1 Hardware Components 47

4.3.2 Required Software Components 49

4.4 Security Plan 50

4.4.1 Security Overview 51

4.4.2 Security Plan 52

5.0 User-Interface 53

5.1 User-Interface Requirements and Constraints 53

5.2 Window/Screen Navigation Diagram 54

5.3 UI Wireframes 56

5.3.1 Home Screen Wireframe 56

5.3.2 Login Screen Wireframe 57

5.3.3 Profile Screen Wireframe 58

5.3.4 Map Screen Wireframe 59

5.3.5 PhotoShoot Scheduler Wireframe 60

5.3.6 Settings Screen Wireframe 60

5.3.7 Reports Screen Wireframe 61

5.4 Reports: “Formal Output” Design 61

6.0 Appendices 64

6.1 Glossary 65

Works Cited 66

1.0 Executive Summary

This document serves as a comprehensive guide for the development of the PhotoMap application, designed to enhance the coordination and planning of photo-shoots between photographers and their clients. The primary audience for this document is the development team, who will use this design document to build and implement the system. The customer for this project includes professional and amateur photographers, models, social media enthusiasts, and businesses seeking photography services.

Project Overview:

PhotoMap, founded by Paulson Kimani, addresses the common challenge of vague or unfamiliar location descriptions in photography assignments. The application integrates a user-friendly map with geolocation features, allowing participants to agree on precise meeting spots by placing pins. This project aims to streamline the logistical aspects of photography, reducing the time and effort spent on location scouting and enhancing the overall planning process for photo-shoots.

Current Progress:

So far, we have defined the problem statement, project vision, scope, and the primary business requirements of the system. We have also identified the stakeholders and their interests, expected costs and benefits, and potential constraints. The feasibility analysis has been conducted, confirming the project's technical, resource, schedule, organizational, legal, and contractual feasibility. Additionally, we have detailed the system's capabilities, non-functional requirements, and design constraints, and outlined the planned system evolution through future versions.

Where the Project is Going:

The next phases of the project involve the detailed design and implementation of the system. This document provides the necessary architectural overview, infrastructure model, hardware and software requirements, and a comprehensive security plan to guide the development team. It also includes user-interface requirements, navigation diagrams, wireframes, and report designs to ensure a consistent and user-friendly application interface.

Document Coverage:

This design document covers the following key areas:

Architecture Overview: A high-level summary of the system architecture, including recommendations for a 3-tier client-server system.

Infrastructure Model: Detailed deployment diagrams showing the major components of the physical system architecture.

Hardware and Software Requirements: A description of the hardware and software components needed to support the application.

Security Plan: A comprehensive plan to address potential security threats and implement necessary controls.

User-Interface Design: Requirements and constraints for the user interface, including navigation diagrams and wireframe mockups for the main screens and reports.

Next Steps:

The next steps for the project include:

Finalizing Detailed Design: Refining the class diagrams, metadata, and component specifications based on feedback.

Development: Implementing the system based on the detailed design provided in this document.

Testing: Conducting thorough testing to ensure functionality, security, and usability.

Deployment: Deploying the application on the recommended infrastructure and preparing for the launch.

User Training and Support: Providing training materials and support to users for a smooth transition to the new system.

By following this comprehensive design document, the development team will be well-equipped to build a robust, secure, and user-friendly PhotoMap application that meets the needs of its diverse user base. The project is on track for a successful launch by the end of the second quarter of 2024, delivering significant benefits to photographers, clients, and other stakeholders involved in the photography planning process.

2.0 Introduction

Product Overview:

PhotoMap is a geolocation-based application designed to enhance the coordination and planning of photo-shoots between photographers and their clients. Founded by Paulson Kimani, the application addresses the common challenge of vague or unfamiliar location descriptions, which often complicate scheduling and logistics in photography assignments. By integrating an interactive map with geolocation features, PhotoMap allows participants to mutually agree on precise meeting spots by placing pins, thus simplifying the site selection process and ensuring both parties are on the same page regarding the location.

Key Requirements:

Geolocation Mapping: Users must be able to select photo shoot locations using an interactive map that supports pin placement, zoom, and rotation.

User Profiles and Social Integration: Profiles should display user activity and link to social media for sharing.

Location Suggestion Feature: A feature to suggest photo shoot locations based on user history using data analytics.

Multi-Platform Compatibility: The app should be accessible on web, iOS, and Android with a consistent interface.

Excluded Features: To maintain focus on professional use, the app will not include messaging, a newsfeed, or an in-app camera.

Stakeholders and Their Interests:

Photographers: Seek to find and connect with clients and discover new photo-shoot locations.

Clients: Want to find and communicate with photographers and explore new locations.

Social Media Users: Interested in sharing and discovering locations, and connecting with others.

Companies: Aim to reach their target audience through sponsored locations and ads.

Development Team: Responsible for designing, developing, and maintaining the application.

Marketing Team: Promotes the application to potential users and advertisers.

Sales Team: Establishes contracts with clients and manages revenue streams.

Researchers: Gather data to improve the application and user experience.

Investors: Look for a return on investment through revenue growth and market share.

General Users: Seek a user-friendly application for finding and sharing locations.

Expected Costs and Benefits:

Development Costs: Include writing the necessary code and performing testing to ensure functionality and user-friendliness.

Marketing Costs: Essential to attract users and gain traction in the competitive app market.

Operational Costs: Include server maintenance and customer support.

Legal Compliance Costs: Involve obtaining licenses and permits to operate legally and ensuring adherence to industry standards.

Security Costs: Include investing in robust security measures to protect user data and ensure privacy.

Constraints:

Availability of Skilled Developers: Limited availability may impact the development timeline and quality. Early recruitment efforts and partnerships are necessary.

Technical Limitations: Varying capabilities across iOS, Android, and web may affect performance. Use of multi-platform development tools and extensive testing is required.

Data Privacy and Security: Handling location data and user information poses significant challenges. Encryption, compliance with data protection regulations, and regular security audits are essential.

Next Steps:

The next phases involve finalizing the detailed design, developing the system, conducting thorough testing, and deploying the application. This document provides the necessary architectural overview, infrastructure model, hardware and software requirements, security plan, and user-interface design to guide the development team through these phases.

For more details, refer to the respective sections of the System Proposal document. This summary serves as a quick reference for developers joining the project or needing a refresher on its key aspects.

2.1 Problem Statement / Project Vision

The project, titled "PhotoMap," founded by Paulson Kimani, is designed to improve the coordination and planning of photo-shoots between photographers and their clients by incorporating a geolocation feature in an interactive map application. This addresses the common challenge of unclear or unfamiliar location descriptions, which often complicate scheduling and logistics in photography assignments. The application simplifies the site selection process by providing a user-friendly map where participants can agree on a precise meeting spot by placing a pin. It ensures that both parties are on the same page regarding the location, making the logistical aspects of photography easier for both professionals and amateurs.

In photography, one of the persistent challenges is the difficulty in communicating and agreeing upon exact locations for photo-shoots. This issue is especially pronounced when descriptions of locations are vague or when the locations are unfamiliar to one or both parties. The resulting confusion can lead to delays and missed opportunities, detracting from the overall experience. The need for a reliable and intuitive solution to this problem is evident, and PhotoMap aims to fill this gap effectively.

PhotoMap aims to become an essential tool for freelance photographers and their clients by transforming how photo-shoot locations are selected and agreed upon. The application aims to provide a seamless, intuitive, and efficient platform that simplifies the logistical aspects of photo-shoot planning. By incorporating geolocation features and an interactive map, PhotoMap ensures that both photographers and clients can easily visualize and agree on precise meeting spots.

The system includes web and mobile applications designed to cater to a wide range of users, including professional and amateur photographers, social media influencers, and businesses seeking photography services. The web application allows users to search for and select locations using an interactive map, check the status of locations, and plan shoots collaboratively. Users can create profiles to showcase their work and connect with others. The mobile application provides all the features of the web application with the added convenience of portability, real-time updates, and integrated navigation tools.

The stakeholders in the PhotoMap project are diverse. Photographers benefit from streamlined planning, improved client coordination, and a platform to discover and share new locations. Clients, including social media enthusiasts and businesses, gain the ability to clearly communicate their location preferences and requirements, reducing misunderstandings and ensuring a smoother photo-shoot experience. Developers and designers benefit from a clear project vision and defined user needs, while investors gain from the potential profitability of the application. Local businesses and property owners can attract photographers and clients to their locations.

The implementation of PhotoMap is expected to yield significant benefits for all stakeholders. Photographers will save time on coordinating locations and enhance client satisfaction. Clients will enjoy clear communication of location details, simplified planning, and an improved overall experience. Developers and investors will benefit from a well-defined roadmap and market potential. Local businesses and property owners will see increased visibility and attraction of photographers and clients to their locations.

PhotoMap aims to launch by the end of the second quarter of 2024, with a development schedule that ensures timely progress. The project timeline includes key milestones such as initial development phases, beta testing, user feedback integration, and final deployment. Market research and user requirements will be conducted in the research and planning phase to define the project scope and create a detailed project plan. The design and prototyping phase will involve developing wireframes and prototypes for the web and mobile applications, conducting usability testing, and refining designs based on feedback. In the development and testing phase, core functionalities will be implemented, geolocation features will be integrated, and comprehensive testing will be conducted. The beta launch and feedback phase will involve launching a beta version of the application to gather user feedback and make necessary adjustments. Finally, the application will be deployed in the final launch phase, and marketing and promotional activities will be initiated to attract users.

In summary, PhotoMap addresses a critical need in the photography community by providing a reliable, intuitive, and efficient platform for planning photo-shoots. The application ensures clear communication and precise coordination between photographers and their clients by leveraging geolocation technology and an interactive map interface. The project's vision and scope encompass a range of functionalities designed to simplify the logistical aspects of photography, catering to a diverse group of users and stakeholders. The anticipated benefits of PhotoMap are substantial, promising improved efficiency, enhanced user experiences, and significant market potential. With a well-defined development plan and a commitment to timely deployment, PhotoMap is poised to become an invaluable tool for freelance photographers and their clients, transforming how photo-shoot locations are selected and agreed upon. Like a compass guiding travelers to their destination, PhotoMap ensures photographers and clients can find the perfect spot with ease and confidence. As the project progresses towards its launch, the team remains dedicated to creating a user-friendly and innovative solution that meets the needs of its diverse user base.

2.2 System Capabilities

1. User Navigates from the Main Menu (ID: 01)

The user opens the application and navigates from the main menu to the map interface to select and pin a location.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

2. User Opens the Map (ID: 02)

The user accesses the map feature to search for and select potential photo-shoot locations.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

3. User Adds a Location (ID: 03)

The user places a pin on the map to indicate a chosen location for the photo-shoot.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

4. Photographer Adds Settings from Camera (ID: 04)

The photographer enters and saves their camera settings in the application for future reference.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

5. User Navigates Out of the Map Feature (ID: 05)

The user exits the map interface after selecting and pinning a location.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

6. Photographer Sends Location to Client (ID: 06)

The photographer securely communicates the chosen location to the client outside of the application.

Refer to Section 4 of the System Proposal and Use Case Descriptions for more details.

2.3 Non-functional Requirements and Design Constraints

1. Security:

The system must ensure the secure handling of user location data and personal information through encryption and compliance with data protection regulations. Regular security audits are required to maintain data integrity. Refer to Section 1.6 for more details.

2. Performance:

The application must perform efficiently, with a responsive interface that loads each webpage within 600 milliseconds. Multi-platform compatibility should ensure consistent performance across web, iOS, and Android devices. Refer to Section 1.6 for more details.

3. Usability:

The user interface must be user-friendly, multilingual, and designed with accessibility features such as screen readers. The simplicity of design is a key theme to ensure ease of use for both professional and amateur photographers. Refer to Section 1.1 and 1.2 for more details.

4. Technical Feasibility:

Existing technologies support the app's development, though the team's learning curve may require additional experts' involvement. The project should leverage multi-platform development tools to handle varying capabilities across devices. Refer to the Feasibility Assessment section in the System Proposal.

5. Legal Compliance:

The system must adhere to legal requirements concerning user data handling and privacy laws. Legal experts should be consulted to ensure full compliance and to avoid potential legal challenges. Refer to Section 1.6 and Feasibility Assessment.

6. Resource Availability:

There are sufficient tools and personnel to start the project, but additional skilled workers may be needed to fill gaps. Early recruitment and partnerships with technology firms specializing in recruiting services are necessary. Refer to Section 1.6 for more details.

7. Schedule Feasibility:

The project timeline avoids busy holiday periods, with a manageable schedule aiming for a launch by the end of the second quarter of 2024. This includes initial development phases, beta testing, user feedback integration, and final deployment. Refer to the Feasibility Assessment section.

8. Organizational Support:

There is strong internal support from departments and senior management, which is crucial for the project's success. This support will help in addressing any technical or legal challenges. Refer to the Feasibility Assessment section.

9. Excluded Features:

The application will not include messaging, a newsfeed, or an in-app camera feature to maintain a focus on professional use. This decision helps streamline the application's functionality. Refer to Section 1.3 for more details.

10. Scalability:

The application should be designed to handle an increasing number of users and data without compromising performance. Scalability is essential for future growth and user satisfaction. This is a new constraint identified during system specification.

For additional details on these requirements and constraints, refer to Sections 1 and 4 of the System Proposal.

2.4 System Evolution

The PhotoMap application will evolve through future developments, expanding on existing functional and non-functional requirements. In Version 1 (MVP), we will deliver the core functionalities essential for the application's launch. These include the interactive map with geolocation pin placement, basic user profiles, and the ability for photographers and clients to agree on precise photo-shoot locations.

Future Version Enhancements:

Version 2: Enhanced Map Features and User Profiles

Enhanced Map Features: Future updates will include advanced map functionalities such as pinch-to-zoom and zoom buttons for better navigation and user experience. Additionally, secure and accurate routing and navigation will be integrated to assist users in reaching selected locations more efficiently.

User Profiles and Post Browser: User profile functionality will be expanded to allow users to display their saved locations and a post browser where photoshoots done at those locations can be securely displayed to potential clients. This enhancement will help photographers showcase their work and attract more clients.

Version 3: Corporate Sponsorships and Additional Features

Corporate and Business Sponsorships: The map interface will include options for corporate or business sponsorships of locations, allowing businesses to promote their spots as ideal photo-shoot locations. This feature will provide an additional revenue stream for the application and create opportunities for businesses to engage with the photography community.

These future developments aim to continuously improve the PhotoMap application, ensuring it meets evolving user needs and market demands. By incrementally adding new features and enhancements, we will maintain user engagement and drive the application's growth.

2.4.1 Version 2 Changes

Use Case Name: User Zooms, Navigates on the App	**ID**: 07	**Importance**: Must Have
Primary Actor: Photographer	**Use Case Type**: Detail, Real	
Supporting Actors: Client		
Stakeholders and Interests:		

Brief Description:

Users can utilize advanced map navigation features, including pinch-to-zoom and zoom buttons, to better explore and select photo-shoot locations.

Trigger:

User interacts with the map interface.

Type (mark one): _X__ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

User opens the map, uses pinch-to-zoom or zoom buttons to navigate, and selects a location.

Sub-flows:

Alternate/Exceptional Flows: N/A
Special Requirements: The map interface must be responsive and support multi-touch gestures.
To do/Issues:

2. User Profile Management (Substantial Change)

Use Case Name: User Manages Profile and Saved Locations	**ID**: 08	**Importance**: Must Have
Primary Actor: Photographer	**Use Case Type**: Detail, Real	
Supporting Actors: Client		
Stakeholders and Interests:		

Brief Description:

Users can manage their profiles, display saved locations, and securely showcase photoshoots done at those locations.

Trigger: User accesses their profile.

Type (mark one): _X__ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

User logs into their account, updates profile information, saves locations, and uploads photos from shoots.

Sub-flows:

User updates profile details, views and manages saved locations, and uploads photos.

Alternate/Exceptional Flows:
Special Requirements: Profile data must be stored securely, and photo uploads should maintain privacy and security.
To do/Issues:

3. Secure Routing and Navigation (New Use Case)

Use Case Name: User Utilizes Secure Navigation to Location	**ID**: 09	**Importance**: Must Have
Primary Actor: Photographer	**Use Case Type**: Detail, Real	
Supporting Actors: Client		
Stakeholders and Interests:		

Brief Description:

Users can access secure and accurate routing to navigate to the selected photo-shoot location.

Trigger:

User selects a location and initiates navigation.

Type (mark one): _X__ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

User selects a location, accesses the routing feature, and follows the directions to the location.

Sub-flows:

User views the route, follows navigation instructions, and reaches the destination.

Alternate/Exceptional Flows:
Special Requirements: Navigation must be accurate and secure, protecting user location data.
To do/Issues:

2.4.2 Version 3 and Beyond Changes

1. Corporate Sponsorships for Locations (New Functional Capability)

Use Case Name: Business Sponsors Location on Map	**ID**: 10	**Importance**: Medium
Primary Actor: Business	**Use Case Type**: Detail, Real	
Supporting Actors: Photographer, Client		
Stakeholders and Interests:		
Brief Description:		

Businesses can sponsor locations on the map, promoting their venues as ideal photo-shoot spots.

Trigger: Business initiates a sponsorship request.

Type (mark one): _X__ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

Business submits a sponsorship request, admin reviews and approves, and the location is highlighted on the map.

Sub-flows:

Business fills out sponsorship details, admin reviews the submission, and the location gets marked as sponsored.

Alternate/Exceptional Flows:
Special Requirements: The sponsorship process must be secure, and sponsored locations should be clearly marked and promoted within the app.
To do/Issues:

2. Advanced Data Analytics for Location Suggestions (New Functional Capability)

Use Case Name: System Suggests Locations Based on User History		**ID**: 11	**Importance**: Medium
Primary Actor: System	**Use Case Type**: Detail, Real		
Supporting Actors: Photographer, Client			
Stakeholders and Interests:			

Brief Description:

The system uses data analytics to suggest photo-shoot locations based on user history and preferences.

Trigger:

User interacts with the app, and the system processes historical data.

Type (mark one): _X__ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

System analyzes user data, generates location suggestions, and displays them to the user.

Sub-flows:

System collects user data, processes it using analytics algorithms, and presents location suggestions.

Alternate/Exceptional Flows:
Special Requirements: Data processing must comply with privacy regulations, and suggestions should be relevant and personalized.
To do/Issues:

3. Enhanced Privacy and Security Features (Substantial Change)

Use Case Name: The app will implement additional privacy measures, such as two-factor authentication and end-to-end encryption for all user data and communications.	**ID**:	**Importance**: High
Primary Actor:	**Use Case Type**:	
Supporting Actors:		

Stakeholders and Interests:

Brief Description:

These enhancements will further protect user data and ensure that all interactions within the app are secure.

Trigger:

Type (mark one): ___ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

Sub-flows:

Alternate/Exceptional Flows:
Special Requirements: Implementation must comply with the latest data protection laws and standards.
To do/Issues:

4. Community and Networking Features (New Functional Capability)

Use Case Name: User Engages in Community and Networking	**ID**: 12	**Importance**: Medium
Primary Actor: Photographer, Client	**Use Case Type**: Detail, Real	
Supporting Actors: Other Users		

Stakeholders and Interests:

Brief Description:

Users can engage in community discussions, join networking groups, and collaborate on photo-shoot projects.

Trigger:

Type (mark one): ___ External ___ Temporal

Relationships:

Association:

Include:

Extend:

Generalization:

The Normal Flow of Events:

User joins groups, participates in discussions, and collaborates on projects.

Sub-flows:

User creates or joins groups, posts messages, and collaborates with others.

Alternate/Exceptional Flows:
Special Requirements: Community features must be moderated to ensure a positive and productive environment.
To do/Issues:

2.5 Document Online

This document is organized into several sections, each covering different aspects of the system proposal and design for the PhotoMap application. Below is an overview of the organization and content of this document:

1.0 Executive Summary

A concise summary of the key points covered in the document, including the problem statement, project vision, system capabilities, and overall conclusions.

2.0 Introduction

2.1 Problem Statement / Project Vision: Details the primary issue PhotoMap aims to address and the overall vision for the project.

2.2 System Capabilities: Lists and describes the functional requirements of the system as outlined in Section 4 of the System Proposal and the Use Case Diagram/Descriptions.

2.3 Non-functional Requirements and Design Constraints: Summarizes the significant non-functional requirements and constraints from Sections 1 and 4 of the System Proposal, including security, performance, usability, technical feasibility, legal compliance, resource availability, and scalability.

2.4 System Evolution: Describes the planned future developments and enhancements for the application.

2.4.1 Version 2 Changes: Lists additional use cases or substantial changes to existing ones for the second version of the application.

2.4.2 Version 3 and Beyond Changes: Includes additional functional capabilities or substantial changes for future versions beyond the MVP.

2.5 Document Outline: Provides an overview of the organization and content of this document.

3.0 Structural Model

3.1 Model Introduction: Introduces the structural models used to represent the system.

3.2 Class Diagrams: Provides class diagrams detailing the structure of the system.

3.3 Metadata: Describes the metadata associated with the system models.

4.0 Architecture Design

4.1 Architecture Overview: Gives an overview of the system architecture.

4.2 Infrastructure Model: Details the infrastructure supporting the system.

4.2.1 Deployment Diagram 1 – Architecture Overview: Provides an overview of the architecture through deployment diagrams.

4.2.2 Deployment Diagram 2 – Nodes and Artifacts: Shows the nodes and artifacts involved in the deployment.

4.3 Hardware and Software Requirements: Lists the hardware and software components required for the system.

4.3.1 Hardware Components: Details the necessary hardware.

4.3.2 Required Software Components: Lists the software components needed.

4.4 Security Plan: Outlines the security measures for the system.

4.4.1 Security Overview: Provides an overview of the security approach.

4.4.2 Security Plan: Details the specific security measures to be implemented.

5.0 User-Interface

5.1 User-Interface Requirements and Constraints: Describes the requirements and constraints for the user interface.

5.2 Window/Screen Navigation Diagram: Provides a diagram showing the navigation flow between windows/screens.

5.3 UI Wireframes: Includes wireframes for the user interface design.

5.4 Reports: “Formal Output” Design: Details the design of formal output reports.

6.0 Appendices

6.1 Glossary: Defines key terms used throughout the document.

6.2 Works Cited: Lists the references and works cited in the document.

6.3 Supporting Documentation: Includes additional supporting documentation relevant to the system proposal and design.

This outline provides a comprehensive structure for understanding the system proposal and design for the PhotoMap application, ensuring all key aspects are covered in a logical and organized manner.

3.0 Structural Model

3.1 Model Introduction

This section provides an in-depth look at the structural model of the PhotoMap application. It includes detailed class diagrams and associated metadata, offering a clear representation of the system's architecture and components. This introduction will guide you through what to expect in the subsequent subsections and how they interrelate.

Section 3.2 Class Diagrams:

This subsection presents the class diagrams, which visually represent the system's structure, showing the classes, their attributes, methods, and the relationships between them. These diagrams provide a blueprint of the application's object-oriented design, illustrating how different parts of the system interact and are organized. Each diagram is crafted to highlight the core functionalities and data structures integral to PhotoMap.

Section 3.3 Metadata:

Following the class diagrams, this subsection delves into the metadata associated with these diagrams. The metadata includes detailed descriptions of each class, attribute, and method outlined in the diagrams. It provides additional context and specifications, ensuring that developers have a thorough understanding of the components and their roles within the system. This detailed metadata serves as a reference for implementing and maintaining the system.

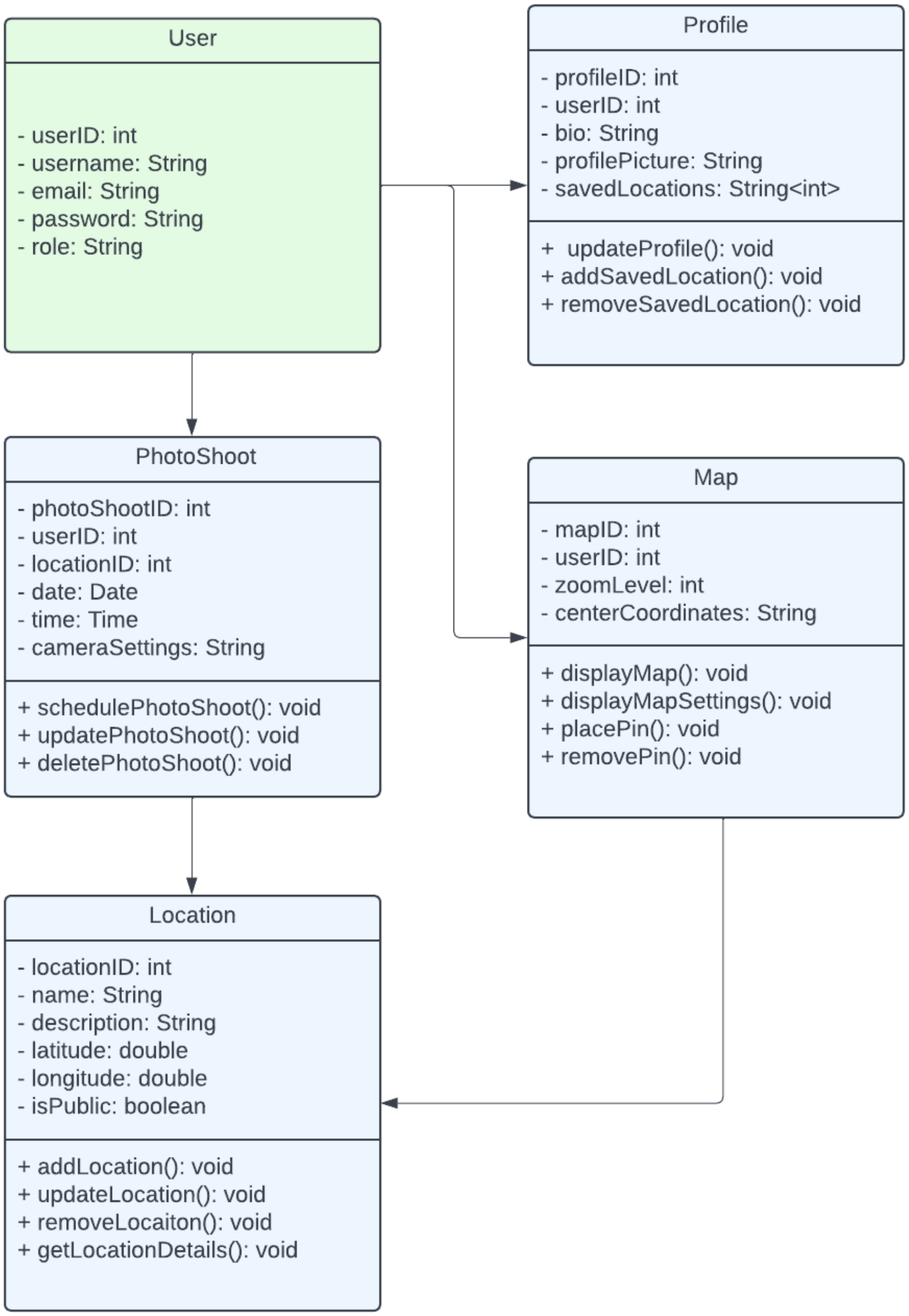
Relation Between Sections B and C:

The class diagrams in Section 3.2 serve as a visual guide to the system's architecture, while the metadata in Section 3.3 offers a textual explanation and detailed specifications of the components shown in the diagrams. Together, these sections provide a comprehensive understanding of the system's structural model. The diagrams give a high-level overview, making it easier to grasp the system's design at a glance, whereas the metadata provides the necessary detail to implement and understand the nuances of each component.

In summary, this section aims to equip developers with both the visual and detailed information required to understand and build the PhotoMap application, ensuring a cohesive and well-documented structural model.

3.2 Class Diagrams

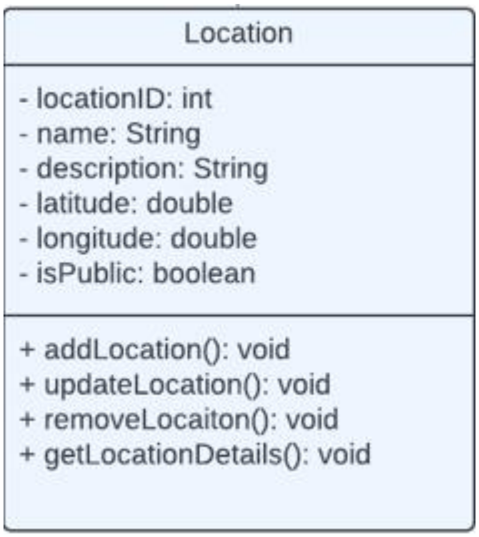
UML Class Diagram



3.3 Metadata

This section provides detailed metadata for each class in the PhotoMap application, including descriptions of the class, its attributes, and methods. The metadata also includes high-level pseudocode outlines for the methods. Each class is listed in alphabetical order, with each class beginning on its own page.

3.3.1 Location



Class Name: Location

Description: Represents a location available for photo shoots, including name, description, and coordinates.

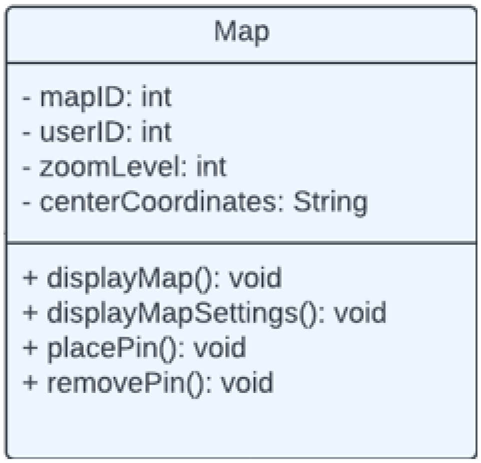
Attributes:

Name	Visibility	Data Type	Default Value	Multiplicity	Description
locationID	Private (-)	int	None	1	Unique identifier for the location.
name	Private (-)	String	None	1	Name of the location.
description	Private (-)	String	None	1	Description of the location.
latitude	Private (-)	double	None	1	Latitude coordinate of the location.
longitude	Private (-)	double	None	1	Longitude coordinate of the location.
isPublic	Private (-)	boolean	True	1	Indicates if the location is public or private.

Name	Visibility	Return Type	Parameters	Description
addLocation()	Public (+)	void	None	Adds a new location to the database.
updateLocation()	Public (+)	void	None	Updates the details of an existing location.
removeLocation()	Public (+)	void	None	Removes a location from the database.
getLocationDetails()	Public (+)	Location	None	Retrieves the details of a specific location.

3.3.2 Map

Map Class Diagram



Class Name: Map

Description: Represents the map interface used in the application for navigating and selecting photo shoot locations.

Attributes:

Name	Visibility	Data Type	Default Value	Multiplicity	Description
mapID	Private (-)	int	None	1	Unique identifier for the map instance.
userID	Private (-)	int	None	1	Identifier for the user associated with the map instance.
zoomLevel	Private (-)	int	1	1	Current zoom level of the map.
centerCoordinates	Private (-)	String	"0,0"	1	Center coordinates of the map.

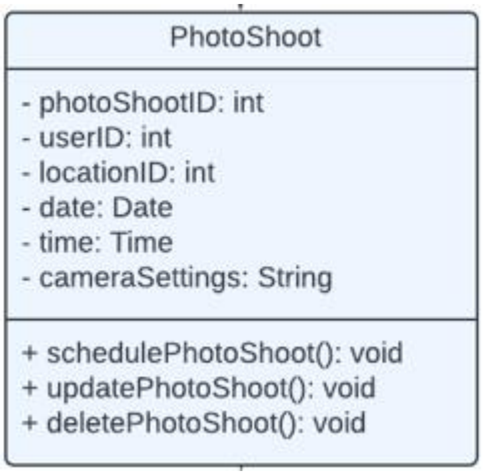
Methods:

Name	Visibility	Return Type	Parameters	Description

displayMap()	Public (+)	void	None	Displays the map to the user.
updateMapSettings()	Public (+)	void	None	Updates the map settings such as zoom level and center coordinates.
placePin()	Public (+)	void	String	Places a pin on the map at the specified coordinates.
removePin()	Public (+)	void	int	Removes a pin from the map.

3.3.3 Photo-Shoot

PhotoShoot Class Diagram



Class Name: PhotoShoot

Description: Represents a scheduled photo shoot, including location, date, time, and camera settings.

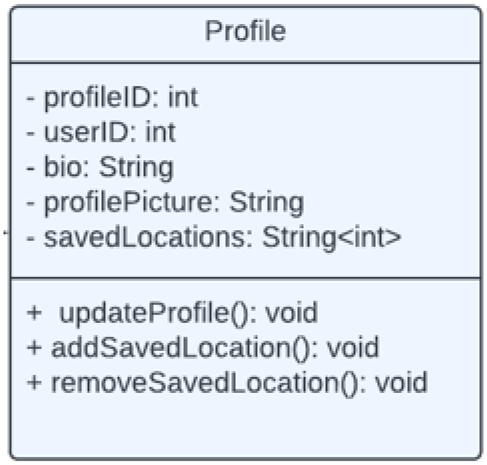
Attributes:

Name	Visibility	Data Type	Default Value	Multiplicity	Description
photoShootID	Private (-)	int	None	1	Unique identifier for the photo shoot.
userID	Private (-)	int	None	1	Identifier for the user who scheduled the shoot.
locationID	Private (-)	int	None	1	Identifier for the location of the shoot.
date	Private (-)	Date	None	1	Date of the photo shoot.
time	Private (-)	Time	None	1	Time of the photo shoot.
cameraSettings	Private (-)	String	None	1	Camera settings used during the photo shoot.

Name	Visibility	Return Type	Parameters	Description
scheduleShoot()	Public (+)	void	None	Schedules a new photo shoot.
updateShoot()	Public (+)	void	None	Updates the details of an existing photo shoot.
deleteShoot():	Public (+)	void	None	Deletes a scheduled photo shoot.

3.3.4 Profile

Profile Class Diagram



Profile

- profileID: int
- userID: int
- bio: String
- profilePicture: String
- savedLocations: String<int>

+ updateProfile(): void
+ addSavedLocation(): void
+ removeSavedLocation(): void

Class Name: Profile

Description: Represents the user's profile, including personal information and saved locations.

Attributes:

Name	Visibility	Data Type	Default Value	Multiplicity	Description
profileID	Private (-)	int	None	1	Unique identifier for the profile.
userID	Private (-)	int	None	1	Identifier for the associated user.
bio	Private (-)	String	None	1	Biography of the user.
profilePicture	Private (-)	String	None	1	URL of the user's profile picture.
savedLocations	Private (-)	List<int>	Empty List	*	List of saved location IDs.

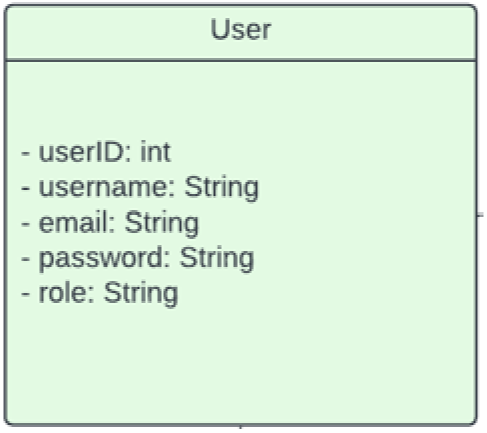
Methods:

Name	Visibility	Return Type	Parameters	Description

updateProfile()	Public (+)	void	None	Updates the user's profile information.
addSavedLocation()	Public (+)	void	int	Adds a new location to the user's saved locations.
removeSavedLocation()	Public (+)	void	int	Removes a location from the user's saved locations.

3.3.5 User

User Class Diagram



Class Name: User

Description: Represents a user of the PhotoMap application, including their basic details and authentication methods.

Attributes:

Name	Visibility	Data Type	Default Value	Multiplicity	Description
userID	Private (-)	int	None	1	Unique identifier for the user.
username	Private (-)	String	None	1	Username chosen by the user.
email	Private (-)	String	None	1	User's email address.
password	Private (-)	String	None	1	User's password for authentication.
role	Private (-)	String	"user"	1	Role of the user (e.g., photographer, client).

Name	Visibility	Return Type	Parameters	Description
login()	Public (+)	boolean	None	Authenticates the user.
logout()	logout()	void	None	Logs out the user.

register()	Public (+)	void	None	Registers a new user.

4.0 Architecture Design

4.1 Architecture Overview

This section provides a high-level overview of the architecture of the PhotoMap application. It is structured to give a clear understanding of the system's infrastructure, design, and requirements. Here, you will learn about the overall architecture, infrastructure model, hardware and software requirements, and the security plan necessary to support and maintain the application.

High-Level Architecture Overview:

The PhotoMap application will be designed as a multi-tier client-server system comprising three main layers: the presentation layer, the application layer, and the data layer. This architecture will facilitate scalability, maintainability, and flexibility, ensuring a robust and efficient system.

Presentation Layer:

This layer includes the user interface components, both web and mobile (iOS and Android), that interact directly with the users. It handles user input and displays the application's output. The presentation layer will communicate with the application layer through RESTful APIs.

Application Layer:

The application layer contains the business logic of the PhotoMap application. It processes user requests, applies business rules, and interacts with the data layer to retrieve and store information. This layer will be hosted on application servers and will include components for user authentication, profile management, map services, and photo-shoot scheduling.

Data Layer:

The data layer is responsible for storing and managing the application's data. It includes databases that store user information, location data, photo-shoot details, and other relevant data. The data layer will be implemented using a relational database management system (RDBMS) to ensure data integrity and security.

Key Architectural Recommendations:

3-Tier Client-Server Architecture: This architecture separates concerns into three layers (presentation, application, and data) to improve scalability and maintainability.

Servers: The system will require the purchase of new servers to host the application and database components. Specifically, two application servers for load balancing and redundancy, and one database server for storing and managing data.

Cloud Services: Utilize cloud services (such as AWS or Azure) to host the application servers and database to ensure scalability, reliability, and cost-efficiency.

Security Measures: Implement robust security measures, including encryption, secure user authentication, and regular security audits, to protect user data and ensure compliance with data protection regulations.

Detailed Section Summaries:

4.2 Infrastructure Model:

This section details the infrastructure required to support the PhotoMap application, including network architecture, server specifications, and deployment diagrams. You will learn about the physical and virtual components needed to deploy and run the application efficiently.

4.3 Hardware and Software Requirements:

This section outlines the hardware and software components necessary for the system. It includes specifications for servers, networking equipment, and the software stack required for development, testing, and production environments. You will learn about the technical requirements and dependencies for setting up the system.

4.4 Security Plan:

This section describes the security measures that will be implemented to protect the system and user data. It covers security policies, data encryption, user authentication, access control, and regular security audits. You will learn about the strategies and best practices to ensure the system's security and compliance with legal standards.

By reading this section, you will gain a comprehensive understanding of the PhotoMap application's architecture, including the infrastructure, hardware and software requirements, and security measures needed to build and maintain a secure, scalable, and efficient system.

4.2 Infrastructure Model

The infrastructure model for the PhotoMap application comprises two deployment diagrams that illustrate the major components of the physical system architecture. These diagrams provide a clear picture of how the system is structured and how its components interact.

4.2.1 Deployment Diagram 1 – Architecture Overview

The Architecture Overview diagram presents a high-level view of the system's infrastructure, showing the primary nodes and their connections. This box-and-line diagram uses consistent icons and node names to depict the physical components of the system.

Architecture Overview Diagram:

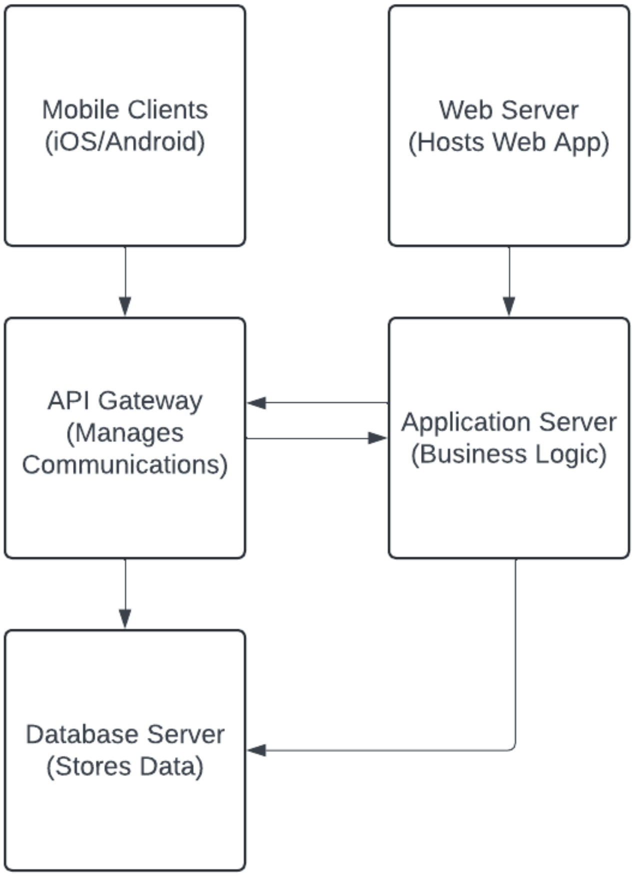

Mobile Clients (iOS/Android)
Web Server (Hosts Web App)
API Gateway (Manages Communications)
Application Server (Business Logic)
Database Server (Stores Data)

Diagram Description:

Web Server (Node): Hosts the web application interface.

Mobile Clients (Node): Represents iOS and Android devices running the mobile app.

Application Server (Node): Handles business logic and processes user requests.

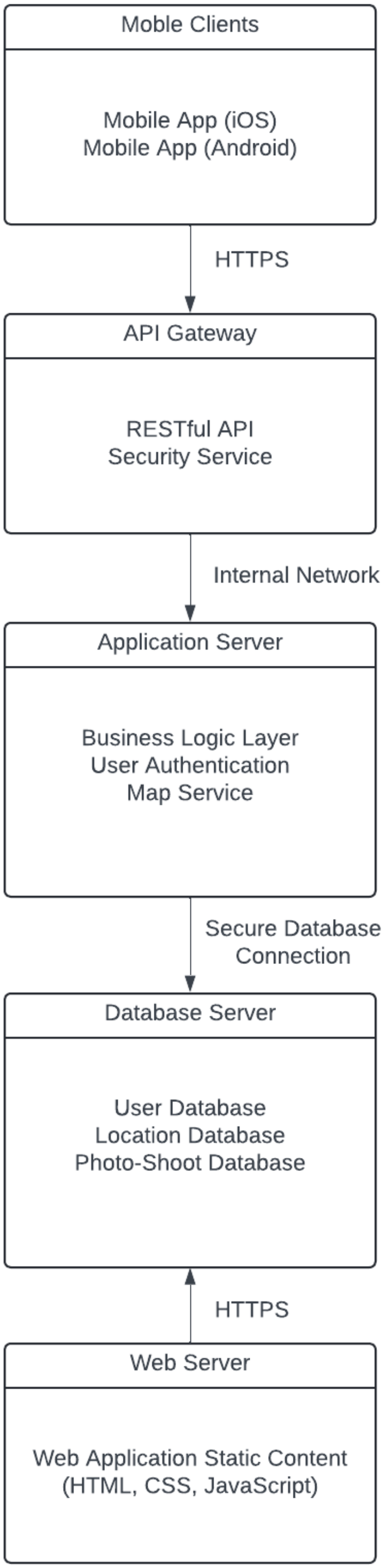
Database Server (Node): Stores user data, location information, and photo-shoot details.

API Gateway (Node): Manages communication between the presentation layer and application layer.

4.2.2 Deployment Diagram 2 – Nodes and Artifacts

The Nodes and Artifacts diagram provides a detailed UML Deployment-with-Component Diagram, showing the specific artifacts deployed on each node and how they connect. This diagram includes components such as web applications, mobile applications, APIs, and databases.

Nodes and Artifacts Diagram:


Moble Clients
Mobile App (iOS)
Mobile App (Android)
HTTPS
API Gateway
RESTful API
Security Service
Internal Network
Application Server
Business Logic Layer
User Authentication
Map Service
Secure Database
Connection
Database Server
User Database
Location Database
Photo-Shoot Database
HTTPS
Web Server
Web Application Static Content
(HTML, CSS, JavaScript)

4.3 Hardware and Software Requirements

This section outlines the hardware and software components required to support the PhotoMap application, including the necessary infrastructure to implement the system's architecture and the software required to make the system operational. The application will use local servers to host all of the data and run the web application.

4.3.1 Hardware Components

Server Hardware:

Web Server:

Specification:

CPU: Intel Xeon E5-2670 or equivalent

RAM: 32 GB

Storage: 1 TB SSD

Network: Gigabit Ethernet

Recommendation: Purchase new server hardware to ensure optimal performance and reliability.

Application Server:

Specification:

CPU: Intel Xeon E5-2680 or equivalent

RAM: 64 GB

Storage: 2 TB SSD

Network: Gigabit Ethernet

Recommendation: Purchase new server hardware, with redundancy for load balancing and failover capabilities.

Database Server:

Specification:

CPU: Intel Xeon E5-2690 or equivalent

RAM: 128 GB

Storage: 4 TB SSD (RAID 10)

Network: Gigabit Ethernet

Recommendation: Purchase new server hardware to handle high I/O operations and ensure data integrity.

API Gateway Server:

Specification:

CPU: Intel Xeon E5-2650 or equivalent

RAM: 32 GB

Storage: 500 GB SSD

Network: Gigabit Ethernet

Recommendation: Purchase new server hardware to ensure secure and efficient API management.

User Devices:

Mobile Devices:

iOS Devices (iPhone, iPad with iOS 13.0 or later)

Android Devices (Android 8.0 or later)

Personal Computers:

Windows 10 or later, macOS 10.14 or later

Alternative Configurations:

Cloud Services: Consider using cloud infrastructure services such as AWS EC2, Azure VMs, or Google Cloud Compute Engine for scalable and flexible deployment.

Hybrid Approach: Utilize a combination of on-premises servers and cloud services to balance cost, scalability, and control.

Recommendation: Given the scalability and flexibility of cloud services, it is recommended to use cloud infrastructure to deploy the application servers and database servers, ensuring high availability and ease of management.

4.3.2 Required Software Components

Operating Systems:

Server OS:

Web Server: Ubuntu Server 20.04 LTS or CentOS 8

Application Server: Ubuntu Server 20.04 LTS or CentOS 8

Database Server: Ubuntu Server 20.04 LTS or CentOS 8

API Gateway Server: Ubuntu Server 20.04 LTS or CentOS 8

User Devices:

Windows: Windows 10 or later

macOS: macOS 10.14 or later

iOS: iOS 13.0 or later

Android: Android 8.0 or later

Database Management Systems:

Primary DBMS: PostgreSQL 13 or later

Alternative DBMS: MySQL 8.0 or later

Web and Application Server Software:

Web Server: Nginx 1.18 or later

Application Server Framework: Node.js 14 or later with Express.js

API Gateway: Kong 2.3 or later or AWS API Gateway

Development Frameworks and Languages:

Frontend (Web): React.js 17 or later

Frontend (Mobile): React Native 0.63 or later

Backend: Node.js 14 or later, Express.js

Mapping Services: Leaflet.js or Google Maps API

Security Software:

Encryption Libraries: OpenSSL 1.1.1 or later

Authentication: OAuth 2.0 or OpenID Connect

Other Software:

Version Control: Git 2.28 or later

CI/CD Tools: Jenkins 2.263 or later, GitHub Actions

Containerization: Docker 19.03 or later

Monitoring and Logging: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana)

Recommendation: Ensure all software components are regularly updated to the latest stable versions to maintain security and performance. Utilize cloud-based services where possible to leverage managed services for databases, API gateways, and serverless functions, reducing the operational overhead.

This comprehensive hardware and software setup will support the efficient development, deployment, and maintenance of the PhotoMap application, ensuring a scalable, secure, and high-performing system.

4.4 Security Plan

4.4.1 Security Overview

The PhotoMap application must address several potential security threats and meet stated security requirements to ensure the protection of user data and the overall integrity of the system. The most significant security concerns include unauthorized access, data breaches, disruptions to service, and ensuring compliance with data protection regulations such as GDPR and CCPA.

Key Security Concerns:

Unauthorized Access: Protecting user accounts and sensitive data from unauthorized access through robust authentication and authorization mechanisms.

Data Breaches: Ensuring that all user data, including location and profile information, is encrypted and securely stored to prevent data breaches.

Service Disruptions: Implementing measures to protect against disruptions caused by cyber-attacks, such as DDoS attacks, and ensuring high availability.

Compliance: Adhering to legal requirements for data protection and privacy, including encryption standards and data handling practices.

Recommended Security Software:

Encryption Libraries: OpenSSL 1.1.1 or later

Authentication: OAuth 2.0 or OpenID Connect

Web Application Firewall (WAF): AWS WAF or equivalent

Intrusion Detection and Prevention Systems (IDPS): Snort or Suricata

Security Information and Event Management (SIEM): Splunk or ELK Stack (Elasticsearch, Logstash, Kibana)

These tools and measures will be used to safeguard the application against various security threats, ensuring the safety and privacy of user data and the reliability of the service.

4.4.2 Security Plan

Threat Component	Unauthorized Access	Data Breaches	DDoS Attacks	Injection Attacks	Physical Theft	Privacy Violations	Server Disruptions	Unauthorized File Access	Man-in-the-Middle Attacks	Code Vulnerabilities
User authenticaiton	1									
Data Storage		2								
Network			3							
Application				4						
Physical Servers					5					
User Data						6				
Operations							7			
File Security								8		
API Communication									9	
Development Process										10

1. Implement OAuth 2.0, multi-factor authentication (MFA), and session management.
2. Use AES-256 encryption for data at rest, regular security audits, and access controls.
3. Deploy AWS WAF, use rate limiting, and employ network firewalls.
4. Validate and sanitize all user inputs, use prepared statements for database access.
5. Secure server locations with access controls, surveillance, and physical barriers.
6. Comply with GDPR and CCPA, use data anonymization and pseudonymization techniques.
7. Implement redundancy, regular backups, and disaster recovery plans.
8. Use file encryption, access control lists (ACLs), and secure file transfer protocols (SFTP).
9. Use HTTPS for all communications, employ TLS 1.2 or higher.
10. Conduct regular code reviews, use static code analysis tools, and provide developer training.

Brief Description:

The security plan for PhotoMap involves a comprehensive approach to mitigate threats across different areas, ensuring robust protection of user data and system integrity. The plan includes implementing OAuth 2.0 for secure user authentication, using AES-256 encryption for data storage, and deploying a web application firewall (WAF) to guard against DDoS attacks.

Physical security measures include securing server locations with access controls and surveillance.

For application-level security, input validation and sanitization will prevent injection attacks, while HTTPS with TLS 1.2 or higher will secure API communications against man-in-the-middle attacks. File security is ensured through encryption and access control lists (ACLs), and a rigorous development process involving regular code reviews and static code analysis will address code vulnerabilities.

By addressing these security aspects, PhotoMap will provide a secure, reliable, and compliant platform for users to plan and coordinate photo-shoots, safeguarding both their data and the overall service.

5.0 User-Interface

5.1 User-Interface Requirements and Constraints

The user interface (UI) of the PhotoMap application is designed with the guiding principles of simplicity, accessibility, and efficiency. The goal is to create an intuitive and user-friendly experience for photographers, clients, and other users who interact with the application. The interface must be easy to navigate, with clear visual cues and consistent design elements across both web and mobile platforms.

Guiding Principles:

Simplicity: The interface should be straightforward, avoiding unnecessary complexity. Users should be able to perform tasks with minimal clicks and clear instructions.

Accessibility: The application must be accessible to users with disabilities. This includes implementing features such as screen readers, high contrast modes, and multilingual support.

Consistency: Consistent design elements and navigation patterns should be used throughout the application to help users familiarize themselves quickly.

Responsiveness: The UI must be responsive, adapting smoothly to different screen sizes and orientations on various devices.

Efficiency: The interface should enable users to complete their tasks efficiently, with quick load times and seamless transitions between screens.

The following sections will detail the various components of the user interface:

5.2. Window/Screen Navigation Diagram: This section provides a diagram that shows how the main screens, forms, and layouts are connected, helping readers understand the flow of the user interface.

5.3. UI Wireframes: This section includes wireframe mockups of the main screens and forms. The wireframes will show the layout and key elements of the user interface, providing a blueprint for the development team.

5.4. Reports: "Formal Output" Design: This section outlines the design of any reports, receipts, invoices, or other formal outputs generated by the application. It will include wireframes for these outputs and describe any print-to-paper options available.

5.2 Window/Screen Navigation Diagram

The Window/Screen Navigation Diagram provides an overview of the main screens and their connections. This diagram helps visualize the flow between different parts of the application.

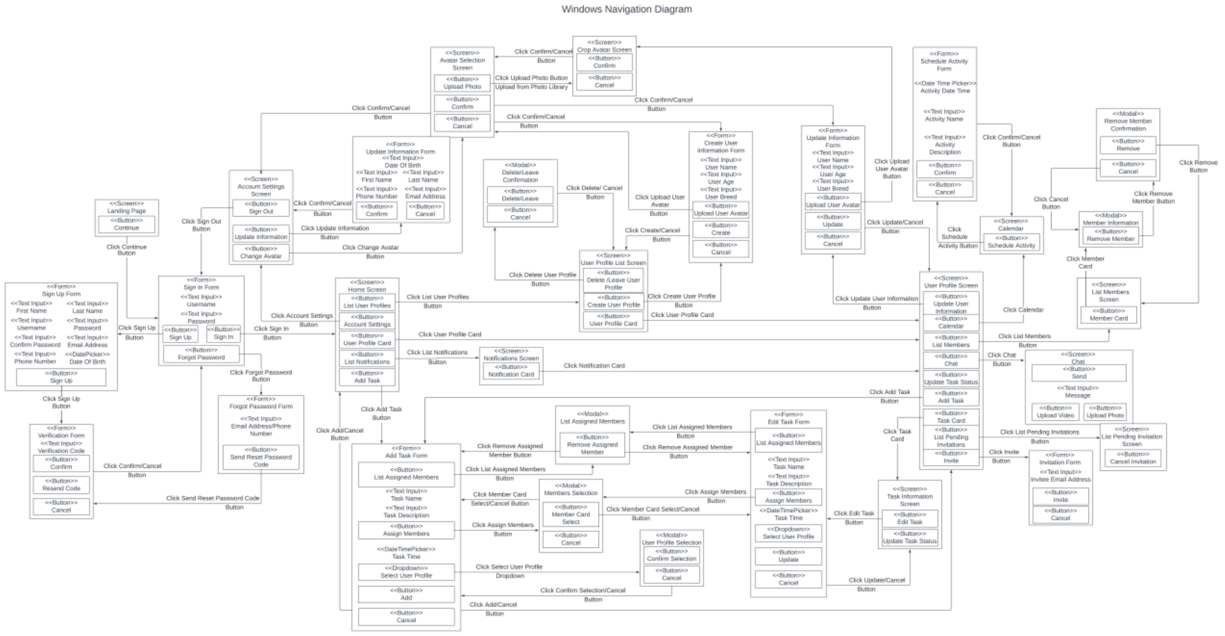

Windows Navigation Diagram
<<Screen>> Landing Page
<<Button>> Continue
Click Continue Button
<<Form>> Sign Up Form
<<Text Input>> First Name
<<Text Input>> Last Name
<<Text Input>> Username
<<Text Input>> Password
<<Text Input>> Confirm Password
<<Text Input>> Email Address
<<Text Input>> Phone Number
<<DatePicker>> Date Of Birth
<<Button>> Sign Up
Click Sign Up Button
<<Form>> Verification Form
<<Text Input>> Verification Code
<<Button>> Confirm
<<Button>> Resend Code
<<Button>> Cancel
Click Confirm/Cancel Button
<<Form>> Sign In Form
<<Text Input>> Username
<<Text Input>> Password
<<Button>> Sign Up
<<Button>> Sign In
<<Button>> Forgot Password
Click Sign Up Button
Click Sign In Button
Click Forgot Password Button
<<Form>> Forgot Password Form
<<Text Input>> Email Address/Phone Number
<<Button>> Send Reset Password Code
Click Send Reset Password Code Button
<<Screen>> Account Settings Screen
<<Button>> Sign Out
<<Button>> Update Information
<<Button>> Change Avatar
Click Sign Out Button
Click Account Settings Button
Click Update Information Button
Click Change Avatar Button
Click Confirm/Cancel Button
<<Form>> Update Information Form
<<Text Input>> Date Of Birth
<<Text Input>> First Name
<<Text Input>> Last Name
<<Text Input>> Phone Number
<<Text Input>> Email Address
<<Button>> Confirm
<<Button>> Cancel
<<Screen>> Avatar Selection Screen
<<Button>> Upload Photo
<<Button>> Confirm
<<Button>> Cancel
Click Confirm/Cancel Button
Click Upload Photo Button Upload from Photo Library
<<Screen>> Crop Avatar Screen
<<Button>> Confirm
<<Button>> Cancel
<<Screen>> Home Screen
<<Button>> List User Profiles
<<Button>> Account Settings
<<Button>> User Profile Card
<<Button>> List Notifications
<<Button>> Add Task
Click List User Profiles Button
Click User Profile Card
Click List Notifications Button
<<Screen>> Notifications Screen
<<Button>> Notification Card
Click Notification Card
<<Modal>> Delete/Leave Confirmation
<<Button>> Delete/Leave
<<Button>> Cancel
Click Delete/ Cancel Button
Click Delete User Profile Button
<<Screen>> User Profile List Screen
<<Button>> Delete /Leave User Profile
<<Button>> Create User Profile
<<Button>> User Profile Card
Click Create User Profile Button
Click User Profile Card
Click Create/Cancel Button
Click Upload User Avatar Button
<<Form>> Create User Information Form
<<Text Input>> User Name
<<Text Input>> User Age
<<Text Input>> User Breed
<<Button>> Upload User Avatar
<<Button>> Create
<<Button>> Cancel
<<Form>> Update Information Form
<<Text Input>> User Name
<<Text Input>> User Age
<<Text Input>> User Breed
<<Button>> Upload User Avatar
<<Button>> Update
<<Button>> Cancel
Click Upload User Avatar Button
Click Update/Cancel Button
Click Update User Information Button
<<Form>> Schedule Activity Form
<<Date Time Picker>> Activity Date Time
<<Text Input>> Activity Name
<<Text Input>> Activity Description
<<Button>> Confirm
<<Button>> Cancel
Click Confirm/Cancel Button
Click Schedule Activity Button
<<Screen>> Calendar
<<Button>> Schedule Activity
Click Calendar
<<Modal>> Remove Member Confirmation
<<Button>> Remove
<<Button>> Cancel
Click Remove Button
Click Cancel Button
Click Remove Member Button
<<Modal>> Member Information
<<Button>> Remove Member
Click Member Card
<<Screen>> List Members Screen
<<Button>> Member Card
Click List Members Button
<<Screen>> User Profile Screen
<<Button>> Update User Information
<<Button>> Calendar
<<Button>> List Members
<<Button>> Chat
<<Button>> Update Task Status
<<Button>> Add Task
<<Button>> Task Card
<<Button>> List Pending Invitations
<<Button>> Invite
Click Chat Button
<<Screen>> Chat
<<Button>> Send
<<Text Input>> Message
<<Button>> Upload Video
<<Button>> Upload Photo
Click List Pending Invitations Button
<<Screen>> List Pending Invitation Screen
<<Button>> Cancel Invitation
Click Invite Button
<<Form>> Invitation Form
<<Text Input>> Invitee Email Address
<<Button>> Invite
<<Button>> Cancel
Click Add Task Button
Click Task Card
<<Screen>> Task Information Screen
<<Button>> Edit Task
<<Button>> Update Task Status
Click Edit Task Button
Click Update/Cancel Button
<<Form>> Edit Task Form
<<Button>> List Assigned Members
<<Text Input>> Task Name
<<Text Input>> Task Description
<<Button>> Assign Members
<<DateTimePicker>> Task Time
<<Dropdown>> Select User Profile
<<Button>> Update
<<Button>> Cancel
Click List Assigned Members Button
Click Remove Assigned Member Button
Click Assign Members Button
Click Member Card Select/Cancel Button
<<Modal>> List Assigned Members
<<Button>> Remove Assigned Member
<<Modal>> Members Selection
<<Button>> Member Card Select
<<Button>> Cancel
<<Modal>> User Profile Selection
<<Button>> Confirm Selection
<<Button>> Cancel
Click Add Task Button
Click Add/Cancel Button
<<Form>> Add Task Form
<<Button>> List Assigned Members
<<Text Input>> Task Name
<<Text Input>> Task Description
<<Button>> Assign Members
<<DateTimePicker>> Task Time
<<Dropdown>> Select User Profile
<<Button>> Add
<<Button>> Cancel
Click Remove Assigned Member Button
Click List Assigned Members Button
Click Member Card Select/Cancel Button
Click Assign Members Button
Click Select User Profile Dropdown
Click Confirm Selection/Cancel Button
Click Add/Cancel Button

Diagram Description:

Home Screen: The starting point for users, providing access to the main features.

Login Screen: Where users can enter their credentials to access the application.

Profile Screen: Displays user profile information and saved locations.

Map Screen: The interactive map for selecting and pinning photo-shoot locations.

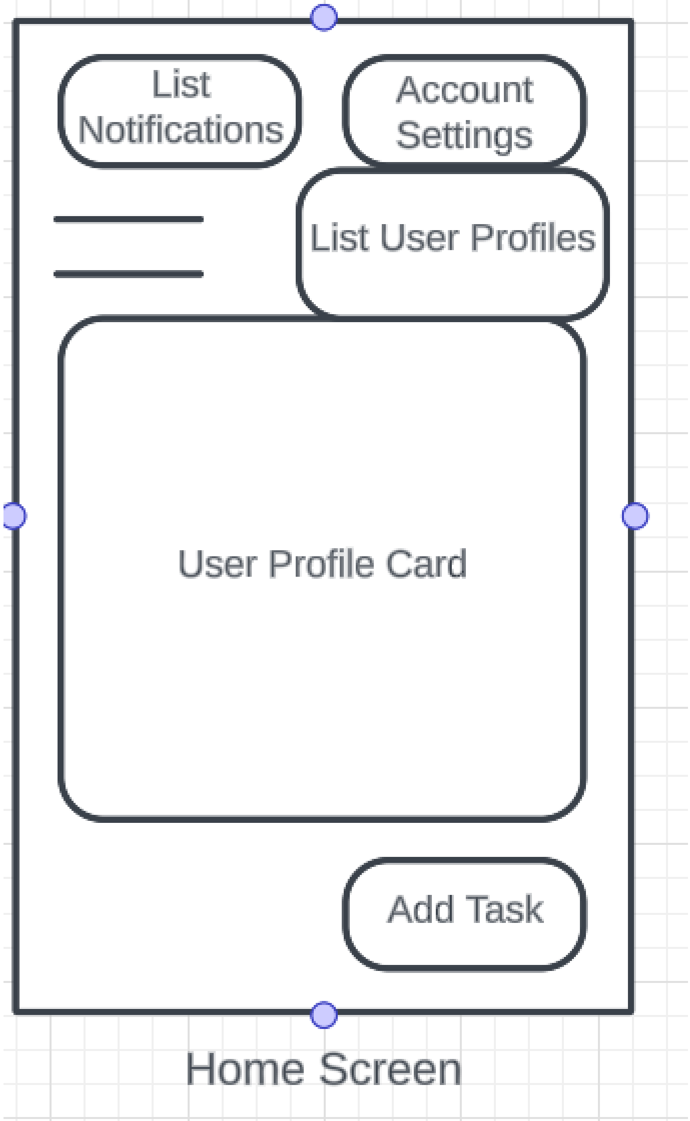
PhotoShoot Scheduler: Allows users to schedule and manage photo-shoots.

Settings Screen: Provides options for configuring application settings.

Reports Screen: Displays generated reports and allows users to print or download them.

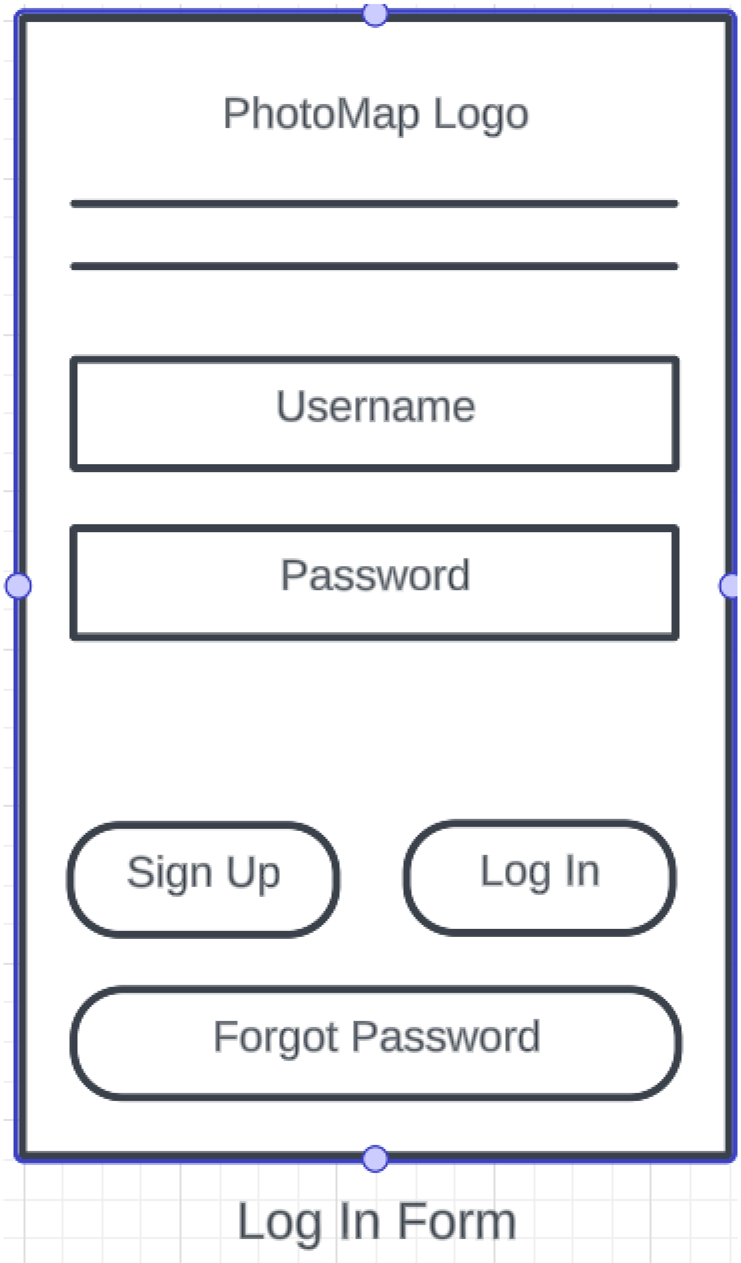
5.3 UI Wireframes

5.3.1 Home Screen Wireframe



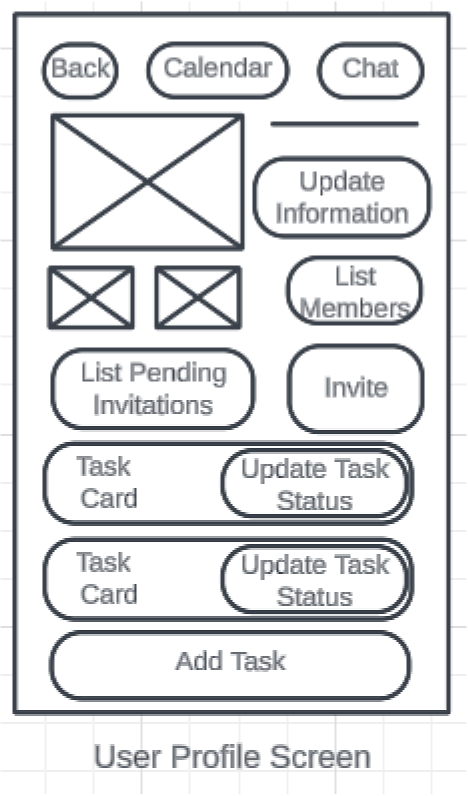
Description: The home screen provides access to login, profile, map, schedule photo-shoot, settings, and reports.

5.3.2 Login Screen Wireframe



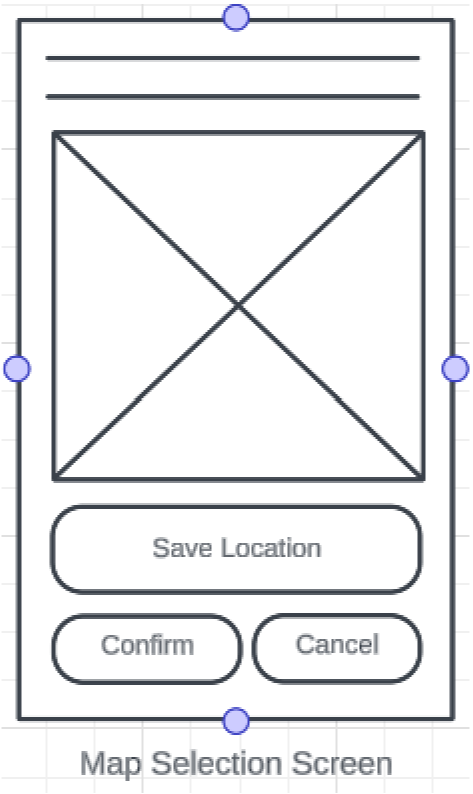
Description: Users can enter their username and password to log in.

5.3.3 Profile Screen Wireframe



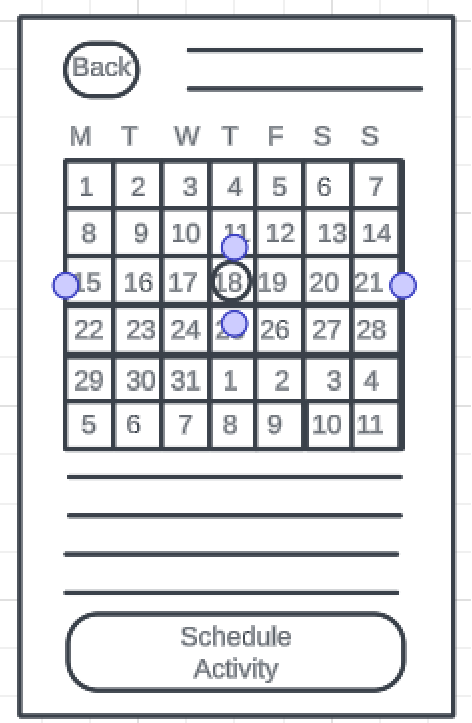
Description: Displays user information, profile picture, bio, and saved locations.

5.3.4 Map Screen Wireframe



Description: Interactive map where users can search for locations, place pins, and view details.

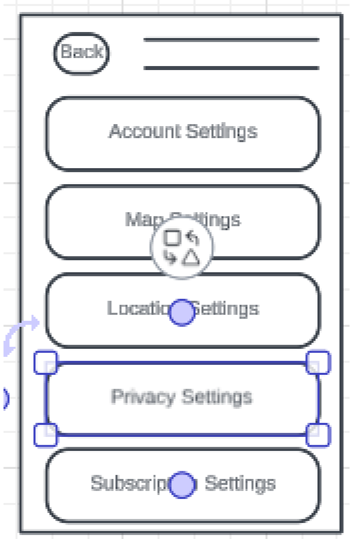
5.3.5 PhotoShoot Scheduler Wireframe



Calendar Screen

Description: Allows users to schedule new photo-shoots, set dates, times, and camera settings.

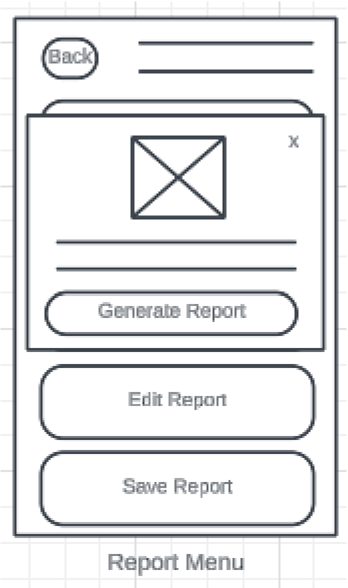
5.3.6 Settings Screen Wireframe



Settings Screen

Description: Provides options to configure application settings such as notifications, privacy, and account details.

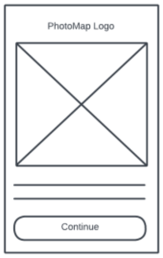
5.3.7 Reports Screen Wireframe



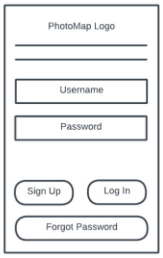
Description: Displays a list of generated reports, with options to view, print, or download.

5.4 Reports: “Formal Output” Design

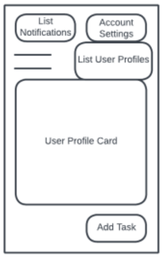
Report Design Wireframe:



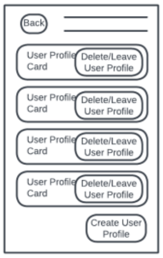
Landing Screen



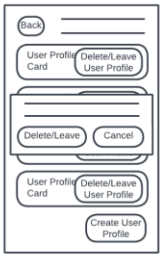
Log In Form



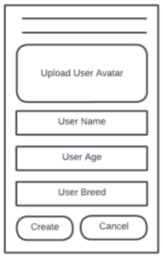
Home Screen


Back
User Profile Card
Delete/Leave User Profile
User Profile Card
Delete/Leave User Profile
User Profile Card
Delete/Leave User Profile
User Profile Card
Delete/Leave User Profile
Create User Profile

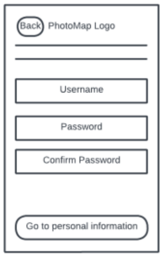
User Profile List Screen



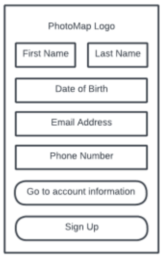
Delete/Leave Confirmation Modal



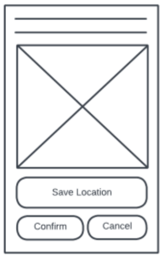
Create User Information Form



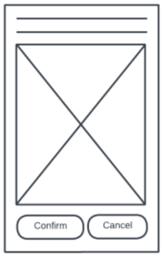
Sign Up Form
(Account Information)



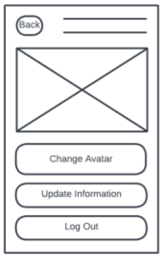
Sign Up Form
(Personal Information)



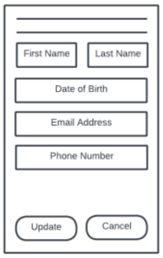
Map Selection Screen



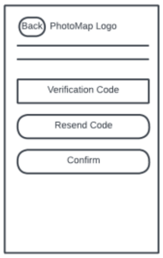
Map Location
Screen



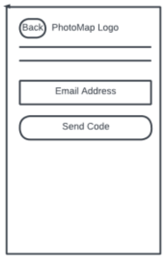
Account Settings Screen



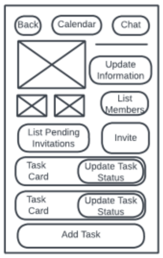
Update Information Form



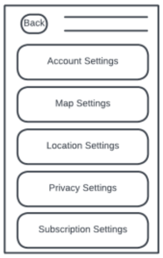
Verification Form



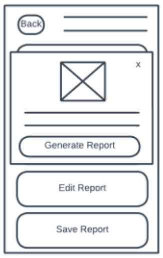
Forgot Password Form



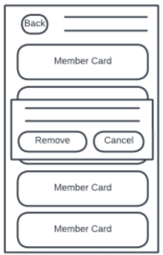
User Profile Screen



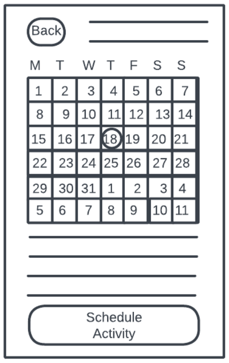
Settings Screen



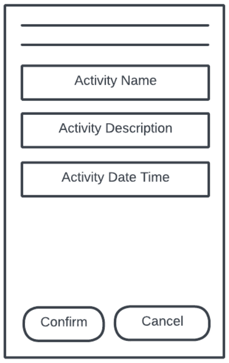
Report Menu



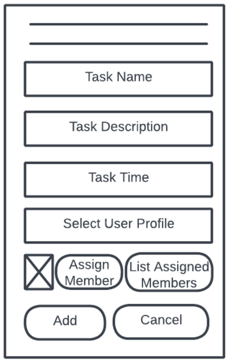
Remove Member
Confirmation Modal



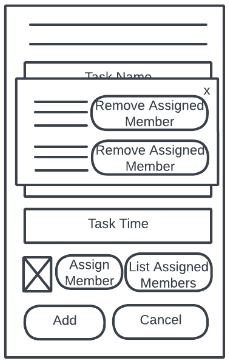
Calendar Screen



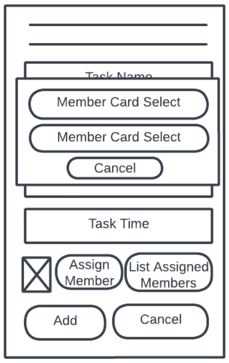
Schedule Activity Form



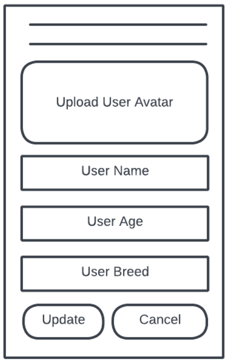
Add Task Form



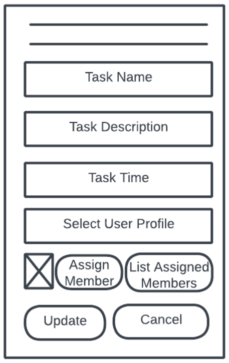
List Assigned Members Modal



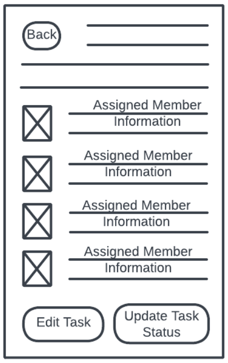
Member Selection Modal



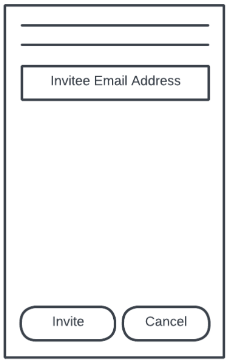
Update User Information Form



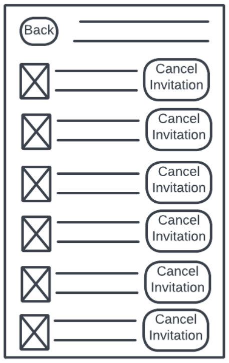
Edit Task Form


Back
Assigned Member Information
Assigned Member Information
Assigned Member Information
Assigned Member Information
Edit Task
Update Task Status

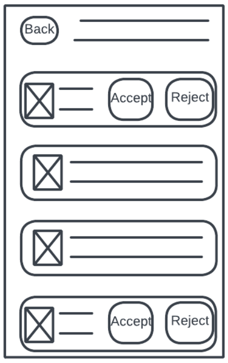
Task Information Screen



Invitation Form



List Pending Invitation Screen



Notifications Screen

Description: This wireframe shows the layout of a typical report generated by the application. It includes the report title, date, user information, and detailed data in a tabular format. Users can print or download the report as a PDF.

Print Options:

Users can print any screen by selecting the "Print" option from the settings menu.

The reports screen specifically allows users to print or download detailed reports.

By providing these detailed wireframes and navigation diagrams, the development team will have a clear understanding of the user interface design, enabling them to implement the necessary components accurately and efficiently.

6.0 Appendices

6.1 Glossary

application – any software that can be used that includes functionalities that are able to be used by the user

communication – the way by which the end user may transmit information between themselves and another user

feasibility – the condition of which a use case is able to be accomplished

geolocation – the location of a place on a map which corresponds to a place on the globe

navigation – the ability for the user to be directed by software securely from the location they are to the location they determine

newsfeed – a page in an application where the user is able to view posts or events that are consecutive

photography – a profession where individuals use digital or film cameras to photograph according to the contract set by themselves and the client

platform – the ways the application can be accessed, either through web browser, iOS mobile operating system, or Android operating system

project initiation request (PIR) – a document whose purpose is to describe the scope of a project.

project – the planning, documentation, and results of work on a task or goal

report – written documentation that describes a certain aspect of the project

security – the means of being certain that the information, means of communication, or otherwise technical qualities are managed to be available to only the necessary parties

social media – any service or platform that allows users to communicate with their friends

specifications – the details of the way a specific product is implemented

stakeholder – anybody who is involved or has investment in a process or project

Works Cited

Cameron, Andy. Systems Design 3150. 2024.

Larman, Craig. Applying UML and Patterns. 2005.