Project Specification

PORTFOLIO TASK 2

Unit code: COS40005
Unit Name: Computing Technology Project A
Submission date: 30/08/2024

Student Name	Student ID	Statement of contribution to the report
Ashim Adhikari	104104333	Hardware Interface, system in context
Nur E Siam	103842784	Problem Statement, High level description, Acceptance Criteria Software Interface, Requirement specification.
Ruffin Remad	103840173	Scope, Stakeholders, Interface Requirements, Documentation
Prabesh Bhattarai	104085535	System Requirement, User Interface, Communication protocol,
Aaradhya Lamsal	103828166	Acceptance Criteria, Documentation, User Interface, Hardware Interface, Software Interface
Ankit Malik	103531273	Hardware Interface, Communication Protocol,
Shivam Sharma	103800575	Acceptance criteria, Functional Requirements, Proof reading

ACKNOWLEDGMENT OF COUNTRY

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campus is in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations. Each team member identifies the Traditional Owners of the land they lived on while completing this work (if living in Australia).

1. PROBLEM STATEMENT

High-Level Summary of the Problem
In Bangladesh, the real estate market faces significant challenges due to the lack of an efficient, reliable, and centralized platform where professionals—such as agents, brokers, and property managers—can effectively showcase their services, and clients can easily find and connect with qualified experts. The existing platform, "Tolet," has not gained significant popularity and is limited by the absence of essential real-time features, such as an interactive map and area-based property search. It does not extend verification to users posting property listings. This can lead to inaccuracies and reduce trust in the platform, making it difficult for users to make informed decisions.

Aim
The aim of this project is to develop a robust mobile and web platform tailored to the needs of the Bangladeshi market, connecting real estate professionals with clients in a secure, reliable, and efficient manner. This platform will address the limitations of "Tolet" by incorporating real-time features, such as an interactive map and area-based search, to enhance user experience and streamline the property search process.

Purpose

The purpose of this project is to create a comprehensive mobile and web application that empowers real estate professionals in Bangladesh to showcase their credentials, manage property listings, and communicate with potential clients. At the same time, it provides users with an intuitive way to discover, evaluate, and connect with qualified professionals. The platform aims to build trust and transparency in real estate transactions by implementing a rigorous verification process for professionals. While the verification process for property listings is currently less stringent, it is recognized as a potential area for future enhancement to improve the platform's overall reliability and accuracy.

Objectives

- **Profile Creation**: Enable real estate professionals in Bangladesh to build detailed profiles that highlight their expertise, experience, contact information, and services, making it easier for users to make informed decisions.

- **Property Listing and Quoting**: Allow proprietors to post property listings and rent-seekers receive quotes from real-estate professionals, fostering competitive pricing and informed decision-making within the local market.

- **Enhanced Property Search:** Enhance the effectiveness of property searches in Bangladesh by incorporating real-time features like an interactive map and property selection based on area. Users will be able to see the locations of properties, pinpoint areas of interest, and customize their search according to their preferences. Provide customers with enhanced search options to assist them in locating properties that better meet their unique needs and preferences.

- **Account Verification**: Use multi-factor authentication (MFA) in addition to CAPTCHA to secure the system. With this two-pronged method, users must prove their human identity with a CAPTCHA test and give several means of identification when logging in, such as a password and a verification code sent to their email or mobile device. MFA and CAPTCHA can guard against phishing attempts, stop unwanted access, and increase user trust.

- **User Posting Flexibility:** Allow proprietors in Bangladesh to freely post property listings, while recognizing the need for future enhancements in verifying these listings to ensure accuracy and reliability.

- **Secure Communication:** Facilitate secure and reliable communication between real estate professionals and clients, ensuring that interactions are both effective and confidential, which is crucial in the developing market context.

This project aims to develop a trusted and user-friendly mobile and web platform specifically designed for the Bangladeshi real estate market. By addressing the limitations of the current "Tolet" platform—particularly the absence of real-time features and the incomplete verification process—this new platform will offer a superior solution that

meets the unique challenges of Bangladesh's real estate sector. The focus on robust verification and real-time search capabilities will ensure a higher standard of service and trustworthiness, essential for building confidence in this developing market.
This version combines the best elements from both drafts, offering a clear, concise, and focused description that addresses both the general challenges of the real estate market and the specific needs of the Bangladeshi context.

2. SCOPE

In-Scope:

1. **Platform Features**: The platform will feature property listings, a bidding system, user profiles, advanced search, and image optimization. These core functionalities, combined with the team's expertise in backend development and efficient image storage, will create a robust and user-friendly platform for property owners and tenants.

    - **Property Listings and Bidding**: Proprietors can post property listings which will allow potential tenants, allowing a bidding system to encourage competitive pricing and ensure users get the best value. The bidding feature is manageable within the project timeframe using existing libraries and frameworks, which the team has prior experience with.

    - **User Profiles:** Users can create profiles to allow rent seekers and landlords to communicate via direct message system.

    - **Search and Filtering**: The app will include search functionality with filters by location and price range. This functionality will be built using the team's familiarity with backend development and relational database management for efficient queries.

    - **Image Compression and Storage**: Upon uploading, images will be automatically compressed and stored in S3 buckets to optimize storage and avoid slowing down the platform. This will also help improve upload speeds and maintain a smoother user experience.

2. **System Design and Architecture**: The knowledge of our team guides the design and architecture of our platform, which is customized to guarantee its performance, scalability, and functionality. To build a reliable and effective system, we've given great thought to the underlying architecture and technological decisions.

    - **Web and Mobile Access:** The platform will provide a responsive web experience in addition to giving priority to mobile application access. This strategy assures accessibility and maximizes resource allocation by utilizing the team's proficiency in web and mobile development.

- **Cloud-Based Deployment:** The app will be hosted using cloud services like AWS (free tier) to leverage scalability without incurring costs. This decision is based on the team's prior knowledge and experience using these platforms for small-scale deployments with limited budgets.

- **API Integration:** The platform will integrate with Google Maps for location-based search. This tool is chosen because the team has worked with it before and is free, suitable for initial deployment.

3. **Development and Maintenance:** Our platform will be built using a minimalist UI/UX design, leveraging tools like Figma and React.js. The backend will be developed with Node.js and Express.js for efficient development and debugging. To streamline the development and deployment process, we will implement a CI/CD pipeline using GitHub Actions. These choices align with our team's expertise and ensure a scalable and maintainable platform.

    - **Frontend Design Principles**: The user interface will follow basic UI/UX design principles such as simplicity, accessibility, and minimalism. Given the team's limited expertise in high-end UI/UX design, a minimalist approach is more achievable within the timeframe. Tools like Figma (for wireframing) and React.js (for frontend development) will be used, aligning with the team's skill set.

    - **Backend Development**: The backend will be developed using Node.js and Express.js, which are familiar to the team. This choice ensures faster development and easier debugging. The backend will handle core functionalities like user authentication and job management.

    - **CI/CD Pipeline:** Basic continuous integration and deployment pipelines using GitHub Actions will be set up to automate testing and deployment. The team's existing knowledge of version control and CI/CD from previous coursework makes this feasible within the project timeline.

4. **Security and Compliance:** To protect user data, we will implement basic encryption. This will help build trust and mitigate legal risks.

    - **Data Encryption:** Basic encryption will be implemented for storing and transmitting sensitive information like login credentials and profile data. The team will rely on libraries they are familiar with, such as bcrypt for password hashing and SSL/TLS for secure data transmission.

5. **Localization:** Localisation includes supporting two languages, Bengali and English, to cater to the diverse linguistic needs of the target market.

- **Multi-language Support**: The app will support both Bengali and English, catering to the primary user base. The team has prior experience implementing multi-language support using i18n libraries, making this feature feasible without introducing complexity.

Out-of-Scope:

1. **Advanced Payment Systems:** The platform will not include digital payment systems like mobile wallets or card transactions due to the market's cash-dominant nature in Bangladesh.

2. **Offline Service Execution and Verification:** While the platform will provide a digital space for property listings, communication, and potential bidding, the actual real estate transactions and in-person verification or inspections will still need to be conducted offline. This means that users will need to meet in person to finalize agreements, conduct property viewings, and complete any necessary paperwork. The platform acts as a digital facilitator but does not directly handle these offline aspects of the real estate process.

3. **Enterprise-Level Infrastructure:** The platform will be scalable, but advanced infrastructure features like auto-scaling, load balancing, and microservices architecture are out of scope. The team's experience is better suited to small-scale, monolithic deployments.

4. **iOS Development:** Only web applications and android support will be available as part of this project.

1. STAKEHOLDER

Primary stakeholders:

- **Property Listers:** Landlords or sellers listing properties (e.g., homes, apartments, land) for rent or sale on the platform. Their primary interest is in reaching potential tenants or buyers efficiently and securely.
- **Property Seekers:** Individuals looking to rent or buy property via the platform. This includes young professionals, students, and families who seek a user-friendly and secure way to find properties that match their criteria.
- **Platform Users:** General users of the platform who may need to communicate, search for properties, and manage listings. This includes both listers and seekers who require secure communication channels and efficient access to property-related services.

Secondary stakeholders:

- **Platform Administrators:** Responsible for managing the day-to-day operations of the platform, including user verification, content moderation, and addressing

user queries. They will also oversee security measures and ensure compliance with data protection regulations.

- **Development Team:** Tasked with designing, developing, and maintaining the platform. This includes frontend and backend developers, UX/UI designers, and database administrators, who will work together to ensure the system's functionality and scalability.
- **Client:** The business owner overseeing the project, ensuring that the platform aligns with business goals and market needs. The client will make key decisions regarding features, design, and future development.
- **Partners and Advertisers:** Entities partnering with the platform for marketing or sponsorship opportunities. These stakeholders may be involved in promoting the platform or leveraging it for business partnerships.
- **Investors and Sponsors:** Financial backers interested in funding the platform to drive growth and innovation. Their focus will be on the platform's scalability, market reach, and potential return on investment.

2. HIGH-LEVEL DESCRIPTION

This overall section prepares the reader, while the following sections present the details.

The real estate market in Bangladesh currently relies on several platforms that serve specific purposes but fall short of delivering a comprehensive, user-friendly experience for all aspects of property transactions. These include:

- **Tolet.com:** A platform focused solely on property leasing. It lacks functionality for buying or selling properties, making it inadequate for users seeking a one-stop solution.
- **Bikroy.com:** A general e-commerce platform that includes property listings among a wide range of products. While it allows buying and selling, its broad scope dilutes the user experience, as it's not specialized in real estate.
- **BD Housing**: Another platform dedicated to property transactions, but it lacks advanced features such as map-based searches, filtering by specific criteria, and dual language support, limiting its usability.
- **Facebook Groups and Marketplace:** Many users resort to these for buying and selling property units, but these are informal, unverified channels, leading to trust issues and inconsistent experiences.

Our new platform is designed to replace and outshine all existing real estate platforms in the Bangladeshi market by providing a comprehensive, specialised, and secure solution for all property transactions—renting, buying, and selling. Unlike platforms such as Tolet, Bikroy.com, BD Housing, and informal Facebook groups, our platform is not merely an upgrade; it is a complete overhaul that integrates best practices and introduces innovative features previously missing in the market.

This platform brings together the functionality of multiple existing platforms under one roof, eliminating the need for users to switch between different services. It offers enhanced security, better user experience, and comprehensive features that address the shortcomings of all current solutions, making it the ultimate go-to platform for real estate transactions in Bangladesh.

Nature of the System

- **New and Complete System**: This platform represents a brand-new development that will replace not only the existing "Tolet" app and website but also aims to surpass other prominent platforms like Bikroy.com, BD Housing, and various Facebook groups and marketplaces. It is built from the ground up using modern technologies specifically tailored to address the diverse needs of the Bangladeshi real estate market. The platform is designed to serve as a comprehensive solution for all types of property transactions—renting, buying, and selling—offering an all-in-one service that the market currently lacks.

- **Replacement of Existing Products:** The new platform replaces all existing real estate platforms in Bangladesh. Providing advanced features such as map-based searches, dual language support (Bangla and English), and rigorous verification processes for professionals and clients will render platforms like Bikroy.com, BD Housing, and Facebook groups obsolete in the real estate domain. The platform's enhanced security measures and user-friendly interface ensure a far superior user experience compared to these older systems.

- **Component of a Larger Ecosystem:** While the platform is primarily a standalone application, it is designed to integrate seamlessly with other essential services, making it a vital component of a broader technological ecosystem. For instance, the platform leverages Google Maps API for location-based searches, which enhances the user experience by allowing property seekers to visualize listings on a map. Additionally, the platform is built on scalable cloud infrastructure (AWS), enabling it to connect with other external services and tools, such as payment gateways and messaging systems, further expanding its functionality and reach.

Key Comparisons and Advantages

1. **Comprehensive Functionality:**
    - **Compared to Tolet:** Unlike Tolet, which only supports leasing, our platform will support leasing, buying, and selling, making it a versatile solution for all users.

    - **Compared to Bikroy.com**: While Bikroy.com is a general marketplace, our platform is specialized in real estate, offering a tailored experience with features like dedicated property searches and professional verification.

- **Compared to BD Housing:** BD Housing lacks advanced search and filtering options. Our platform will provide map-based searches, detailed filtering options, and a more user-friendly interface with dual-language support.

- **Compared to Facebook Groups/Marketplace:** Our platform provides a formal, secure, and verified environment, reducing the risks associated with informal channels.

2. **Enhanced User Experience:**
   - **Dual Language Support:** The platform will be available in both Bangla and English, making it accessible to a wider audience, which is particularly useful in a diverse market like Bangladesh.

   - **Map-Based Search:** Users will be able to search for properties using an interactive map, a feature that is currently missing from BD Housing and other platforms.

   - **Professional and Client Verification**: The platform will include verification processes for both professionals and clients, ensuring that all listings are trustworthy and reducing the risk of fraud, which is common on less formal platforms like Facebook groups.

3. **Advanced Features:**
   - **User Profiles and Management:** Both property listers and seekers will have customizable profiles, allowing them to manage their listings and interests more effectively.

   - **Notifications and Updates:** Real-time notifications will keep users informed about new listings, updates, and messages, ensuring they never miss an opportunity.

   - **Integrated Communication Tools:** Secure messaging and communication tools will allow users to interact directly through the platform, streamlining negotiations and transactions.

System Diagram Description

1. **User Interfaces:**
   - **Mobile App:** Used by users on Android.
   - **Web App:** Used by users on web browsers.

2. **Backend and Services:**

- **AWS Amplify:** This service will host both the mobile and web applications, providing continuous integration and deployment.

- **AWS Lambda:** Handles backend logic, including API requests and interactions with other services like databases and external APIs.

- **Amazon RDS (MySQL Database):** Stores user profiles, property listings, and transactional data securely.

- **Amazon S3:** Stores media files such as property photos, ensuring they are easily accessible and scalable.

- **Amazon SES:** Manages the sending of email notifications to users, such as alerts for new listings or messages.

- **Google Maps API:** Integrated for location-based search, allowing users to find properties using an interactive map

Connections

- **Mobile App and Web App:** Both connect to AWS Amplify for hosting and CI/CD.

- **AWS Amplify:** Connects to AWS Lambda for executing backend logic.

- **AWS Lambda:** Interacts with Amazon RDS for database queries, Amazon S3 for media storage, Amazon SES for email services, and Google Maps API for location-based searches.

Key Points of Interaction

- The Mobile App and Web App serve as the primary interfaces for users.

- AWS Amplify acts as the central point for hosting, deployment, and initial API routing.

- AWS Lambda handles the core logic, connecting to various services to manage data, files, notifications, and search functionality.

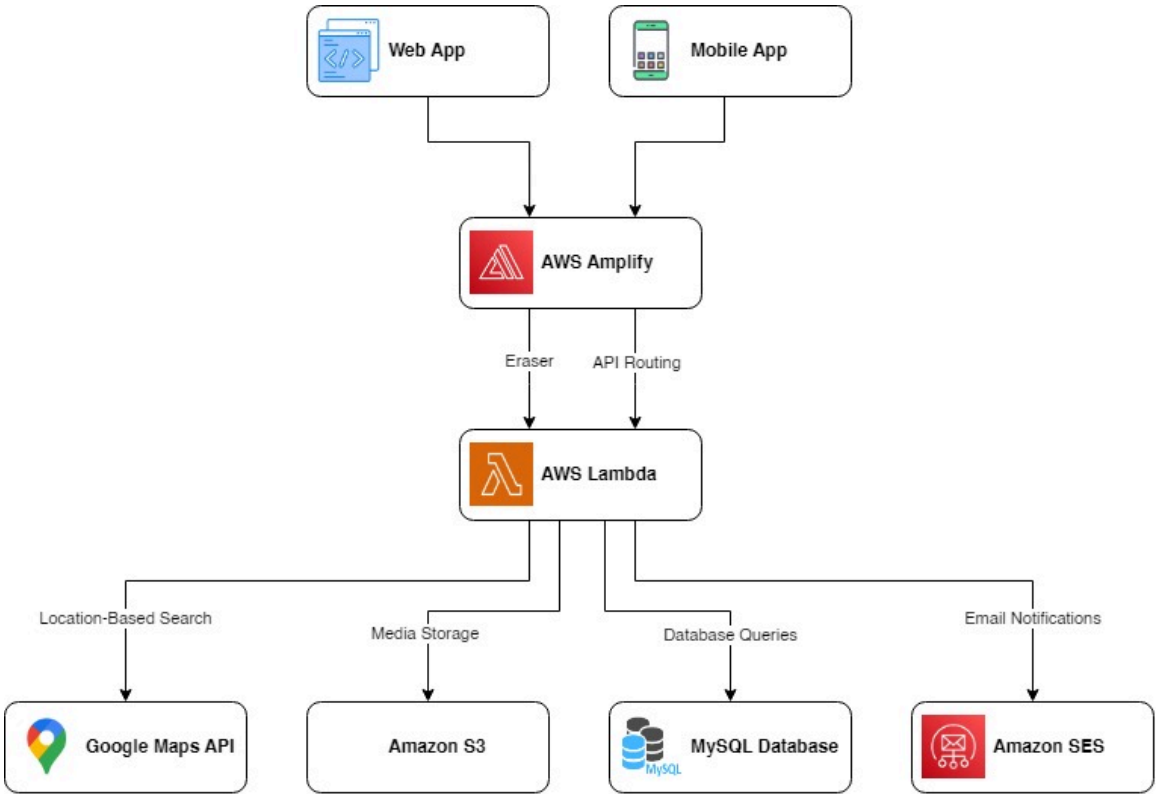

Fig1. software to system interaction

This architecture ensures scalability, security, and an enhanced user experience, integrating all the advanced features necessary for a comprehensive real estate platform.

A. PRODUCT FEATURE

1. Property Listings

- **Multi-Property Type Support:** A wide range of property types that include apartments, lands, and agricultural land. These various property attributes will require flexible listing templates and search filters.

- **Area-Based Listing:** Sales listings of properties by area basis. Implement a robust location-based search/filtering system. It may also require mapping services to facilitate visualization.

- **Photo Uploads**: Allow Users to post multiple photos of properties. Development of an end-user-friendly photo upload system with compression and optimization for better, faster loading. Maybe even a virtual tour feature in future iterations.

- **Price Range Suggestions:** Suggesting appropriate price ranges for listings. It would involve developing an algorithm that analyses similar properties in the area to suggest price ranges.

2. User Interaction and Communication

- **Booking Inspections:** Users can schedule viewings for properties through the platform. A calendar integration system for property listers and a booking system for seekers with reminders and confirmation messages are to be developed.

- **Privacy Oriented Communication:** A secure system to share contact information mainly by email / phone. Develop a cloaked system of communications that enables users to connect for business without directly exposing their personal contact information. As much as possible, implement temporary phone numbers for calls.

3. User Experience and Accessibility:

- **Dual User Interface:** The site caters to listers of properties and rent-seekers. Two user journeys must be designed with clear navigation between them.

- **Localization:** The platform will be fully localized in both Bengali and English. This includes not just translation, but also cultural nuances embedded in property descriptions and user interactions, ensuring accessibility for Bengali-speaking user.

4. Security and Verification

- **Robot Check:** Ensure the implementation of a functional CAPTCHA or similar systems, which prevent automated abuse effectively. Include an intuitive, yet most effective bot prevention system, particularly on listing creation and high-volume actions.

5. Search and Discovery

- **Advanced Filters:** An all-inclusive set of filters on search that meet all characteristics of a property. Allowing this filter system to be flexible and intuitive for the user, it would then allow them to filter through all the available properties in terms of location, price range, type of property, and facilities provided.

- **Map-Based Search:** Visual geographic representation of property listings. This requires an integrated mapping service to give an interactive, map-based search interface which will provide users with the ability to know visually which properties are in which areas.

6. **Trust and Transparency:**

- **User Reviews and Ratings:** Allow every user to rate and review a property/lister after some interaction with them. This would entail developing a review system with suitable checks against spamming and other forms of misuse.

- **Listing Verification:** Verify the authenticity of high value/frequently booked properties. Develop a verification procedure; this could entail in-person checks or document verification for certain properties.

4.2 SYSTEM REQUIREMENTS

Development Environment

Software:

- **Operating System**: Windows, macOS, or Linux
  Flexibility in OS choice based on team preferences and compatibility with development tools.

- **Development Tools**: Visual Studio Code, Git with GitHub
  Visual Studio Code supports a range of programming languages and features needed for development, while Git with GitHub facilitates version control and team collaboration.

- **Frameworks/Libraries**: Node.js for backend, React.js for frontend
  Node.js and React.js are chosen for their ease of use and familiarity with the team, allowing for efficient development and integration.

Production Environment

- **Operating System:** Linux-based OS (e.g., Ubuntu Server) or potentially AWS Managed Services Linux is preferred for its stability and security. However, we might opt for AWS-managed services like AWS Elastic Beanstalk or AWS Lambda, depending on technical constraints and team capability.

- **Web Server:** AWS EC2 with Nginx or Apache
  Nginx or Apache are both reliable choices for serving web applications. We'll use AWS EC2 instances, which can be configured with either server based on final needs and requirements.

- **Database:** AWS RDS (Relational Database Service) for MySQL
  MySQL managed by AWS RDS offers a robust solution with features like automatic backups and scaling, simplifying data management and maintenance.

- **Hardware:** Processor: AWS EC2 instances with suitable instance types (e.g., t3.micro or t3.small for initial deployment)
  Provides adequate processing power for handling server tasks efficiently.

- **RAM:** AWS EC2 instances with 4 GB of RAM
  Sufficient for managing server operations while balancing cost and performance.

- **Storage:** AWS EBS (Elastic Block Store) with 50 GB SSD
  SSD ensures quick data access, and AWS EBS provides scalable storage options as needed.

- **Networking:** Internet Connection with High-bandwidth, low-latency
  Ensures efficient handling of web traffic and a smooth user experience.

- **Backup and Recovery:** Automated Backups like AWS RDS automated backups and snapshots
  AWS RDS takes care of database backups, and EC2 snapshots are used for server data, ensuring quick recovery and data protection.

- **Security:** Basic Measures like AWS Security Groups, Network ACLs, and regular updates
  AWS Security Groups and Network ACLs help protect against unauthorized access. Regular updates and patches are essential to keep the system secure.

4.3 ACCEPTANCE CRITERIA

1. **User Experience (UX)**
   - **Criteria:** The system should provide an intuitive, user-friendly interface that is accessible and easy to navigate for all users, including property listers, seekers, and real estate professionals.
   - **Rationale:** A positive user experience is crucial for user adoption and satisfaction. The interface should minimize the learning curve and facilitate seamless interactions, encouraging frequent and effective use of the platform.

2. **System Performance**
   - **Criteria:** The system should operate quickly and effectively, managing large amounts of data and user traffic without experiencing any noticeable hiccups or delays. To preserve responsiveness and stability in a variety of settings and devices, it should also optimize resource utilization.

   - **Rationale:** Maintaining user happiness requires high performance because slow or lagging systems can irritate users and reduce usage. All users will experience a seamless system thanks to effective resource management, which keeps the system responsive and stable even under high load.

3. **Security and Data Protection**

- **Criteria:** The system must implement robust security measures, including data encryption (e.g., using secure protocols like HTTPS with TLS), and require strong authentication methods such as two-factor authentication (2FA) for users, especially for administrative access.

- **Rationale:** Robust security measures are necessary to stop data leaks and safeguard private user data, which builds system dependability and confidence. Adherence to data protection standards serves to safeguard not just the privacy of users but also the organization's reputation and legal ramifications.

4. **Scalability**
   - **Criteria:** The system should be designed to support growth, with the ability to scale horizontally, meaning the system can handle increased load by adding more servers or resources into the system without significant changes to the application.

   - **Rationale:** Scalability ensures that the platform can accommodate a growing user base and increasing amounts of data without degrading performance, supporting the platform's long-term success.

5. **Reliability**
   - **Criteria:** The system must have a high degree of dependability, little downtime, and rapid failure recovery. To guarantee ongoing operation and data integrity, redundancy measures and routine maintenance inspections should be incorporated.

   - **Rationale:** Building user trust and making sure the system is always available when needed depend heavily on reliability. The system can prevent disruptions and offer a dependable service that customers can rely on by including redundancy and proactive maintenance. This is especially crucial for applications that involve critical or time-sensitive operations.

6. **Compliance**
   - **Criteria:** The platform must comply with local data protection laws in Bangladesh, such as the Digital Security Act, and adhere to basic international best practices for software development and data security.

   - **Rationale:** Compliance with legal and regulatory requirements is necessary to avoid penalties, maintain user trust, and ensure the system's legitimacy.

7. **Maintainability**

- **Criteria:** The system's codebase should be well-documented and follow best practices for readability and modularity. Regular updates and bug fixes should be supported by a clear process.

- **Rationale:** Maintainability is crucial for the efficient development and deployment of updates, ensuring that the platform can evolve over time without introducing significant new bugs.

8. **User Feedback Mechanism**
   - **Criteria:** The system should allow users to leave feedback or report issues regarding property listings or professional interactions via email, which will be responded to by the administrator.

   - **Rationale:** A feedback mechanism is essential for continuous improvement, allowing the development team to address issues and enhance the platform based on user input.

These acceptance criteria form the foundation for evaluating the success of the system. Meeting these criteria will ensure that the platform is functional, secure, and user-friendly, thereby fulfilling its intended purpose and delivering value to all stakeholders.

4.4 DOCUMENTATION

1) **User Guides**
   - **Property Lister:** A step-by-step guide to creating your property listing, managing it, and finally optimizing it with the right pricing, photo upload, and inquiry responses.

   - **Property Seeker:** A walkthrough for renters on how to find a property, apply filters, contact listers, and book inspections.

   - **Administrator's Manual:** A detailed guide for administrators of the platform regarding the management of users, content moderation, and handling the overall operations of the platform.

2) **Technical documentation**
   - **System Architecture Overview:** Detailed documentation regarding the structure of the platform, including back-end services, databases, client applications, and cloud services which is Amazon Web Services (AWS).

   - **Database Schema and Data Dictionary:** Full documentation of database structure, relationships between them, and definitions of data.

- **API Documentation:** Detailed documentation of all API endpoints, formats of request and response, and methods of authentication.

- **Third-Party Integration Guide:** Steps for integrating with external services like mapping APIs, payment gateways, and email services.

3) **Server Setup and Configuration Guide:**

- **Mobile App Deployment Guide (Android):** Step-by-step process to deploy the mobile apps to respective app stores.

- **Web Application Deployment Guide:** The directions on the deployment and update of the web application.

4) **System Maintenance Guide**:
   - **Troubleshooting Manual:** Common issues and their solutions, diagnostic procedures, and escalation protocols.

   - **Backup and Recovery Procedures:** Step-by-step procedures to back up data and recover from various failure scenarios.

5) **Security Documentation:**
   - **Overview of Security Features:** Detailed description of all the security measures taken concerning the platform.

   - **Data Privacy and Protection Guidelines:** Comprehensive guidelines on the protection of the users' data, explaining the methods of encryption used and how access is controlled.

   - **Compliance Documentation:** Information on how the platform complies with Bangladeshi regulations concerning data protection and real estate.

6) **Quality Assurance:**
   - **Test Plans and Test Cases:** Full set of test scenarios to include all functionalities, edge cases, and any other related topics.

   - **Quality Assurance Procedures:** Detailed procedures that must be followed to ensure the quality and reliability of each release.

7) **Developer Documentation:**
   - **Code Documentation:** Comments inline, and external documentation explaining how the structure of the codebase is, and which key algorithms have been used.

- **Development Environment Setup Guide:** A guide on setting up a local development environment.

- **Coding Standards and Best Practices:** Best practices for the quality and consistency of the code.

8) **Legal Documents**:
   - **Terms of Service:** Clear explanation of the rules and regulations for using the platform.

   - **Privacy Policy:** Detailed information on how user data is collected, used, and protected.

   - **End-User License Agreement (EULA):** Legal agreement between the platform and its users.

   - **Translation Guidelines:** Instructions for translating the platform into Bengali, including cultural considerations.

   - **Cultural Considerations for Content:** Guide to ensuring all content is culturally appropriate for the Bangladeshi market.

9) **User Feedback and Support**:
   - **User Feedback Collection Procedures:** Methods for gathering and analysing user feedback to improve the platform via email.

   - **Support Ticket System Guide:** Instructions for managing user support requests efficiently.

3. REQUIREMENT SPECIFICATION

The purpose of this section is to clearly outline the key requirements necessary for the development and implementation of the real estate platform. These requirements are categorized into major features of the system. Each requirement has been carefully identified and documented to ensure the system meets the expectations of all stakeholders, providing a seamless user experience and robust functionality.

The detailed requirements are structured to include a unique identifier for easy reference, a clear statement of the requirement, the rationale behind it, and any additional notes if applicable. This structured approach ensures that all aspects of the system are comprehensively covered, leaving no room for ambiguity during the development process.

User Account Management

1) **User Registration and Login**
    - Requirement ID: UAM-001
    - Requirement Statement: The platform shall allow users to register and log in using their personal information, such as name, email, phone number, and password. Moreover, users will be able to reset password as well.
    - Rationale: This is the foundational feature that enables users to create an account and access personalized services.
    - Note: The system should support social media login options, such as Google and Facebook, to streamline the registration process. Only one account can be created on a single email and phone number.

2) **Profile Management**
    - Requirement ID: UAM-002
    - Requirement Statement: Users shall be able to update their profiles, including personal details, contact information, and preferences.
    - Rationale: Profile management allows users to keep their information up to date, which is crucial for personalized recommendations and communication.
    - Note: The profile should include verification status for professionals.

Property Listing Management

1) **Property Listings**
    - Requirement ID: PLM-001
    - Requirement Statement: The platform shall allow users to create and manage property listings with details such as property type, location, price, and images.
    - Rationale: This is the core functionality of the platform, enabling users to list their properties for sale, rent, or lease.
    - Note: Listings should be easily searchable and filterable based on user preferences.

2) **Advanced Search and Filtering**
    - Requirement ID: PLM-002
    - Requirement Statement: The platform shall provide advanced search and filtering options, allowing users to search for properties by criteria such as location, price range, property type, and amenities.
    - Rationale: Advanced filtering enhances the user experience by making it easier to find suitable properties.
    - Note: Filters should be intuitive, easy to apply, and clearly labelled.

Communication and Interaction

1) **Integrated Messaging System**

- Requirement ID: CI-001
- Requirement Statement: The platform shall include an integrated messaging system that allows users to communicate directly through the platform.
- Rationale: Direct communication between users and real estate professionals is essential for negotiating deals and addressing queries.
- Note: Messages should be encrypted to ensure privacy and security.

2) Notifications

- Requirement ID: CI-002
- Requirement Statement: The platform shall provide real-time notifications for messages, new listings, updates, and other relevant events.
- Rationale: Notifications keep users engaged and informed, ensuring they do not miss important opportunities.
- Note: Users should have the option to customize their notification preferences.

Real-Time Features

1) Real-Time Updates

- Requirement ID: RTF-001
- Requirement Statement: The platform shall support real-time updates for property listings, searches, and user interactions.
- Rationale: Real-time updates ensure that users have access to the most current information, improving the overall user experience.
- Note: This feature should be optimized to minimize latency and ensure quick loading times.

Security and Data Protection

1) Data Encryption

- Requirement ID: SDP-001
- Requirement Statement: The platform shall implement robust data encryption for both data at rest and in transit, using industry-standard encryption methods such as AES-256.
- Rationale: Protecting user data from unauthorized access is crucial for maintaining trust in the platform.
- Note: Regular security audits should be conducted to ensure the continued security of the platform.

Administrative Functions

1) Admin Dashboard

- Requirement ID: AF-001
- Requirement Statement: The platform shall include an admin dashboard that allows administrators to manage users, properties, and platform settings.
- Rationale: An admin dashboard is essential for maintaining the platform, managing content, and addressing user issues.

- Note: The dashboard should include analytics and reporting features to track platform performance.

2) Content Moderation

- Requirement ID: AF-002
- Requirement Statement: The platform shall include tools for content moderation, allowing administrators to review and approve property listings and user-generated content.
- Rationale: Content moderation is necessary to ensure that the platform remains trustworthy and free of inappropriate content.
- Note: Moderation tools should be user-friendly and efficient.

User Feedback and Support

1) Feedback System

- Requirement ID: UFS-001
- Requirement Statement: The platform shall include email-based feedback system that allows users to submit feedback and report issues.
- Rationale: Collecting user feedback is important for continuous improvement of the platform.
- Note: Feedback should be categorized and tracked to ensure timely responses and resolutions.

2) Support and Help Centre

- Requirement ID: UFS-002
- Requirement Statement: The platform shall include email-based support and help centre that provides users with resources, FAQs, and contact options for support.
- Rationale: A well-organized support system improves user satisfaction and helps resolve issues quickly.
- Note: The help centre should be easily accessible from the platform's main interface.

5.1.FUNCTIONAL REQUIREMENTS

User Account Management

1) User Registration

- **Requirement:** The system should allow users to register on the platform by providing their personal information, including name, email address, phone number, and password.
- **Rationale:** Enables users to create personalized accounts for interacting with the platform.
- **Note:** Only one account is permissible per phone number to prevent duplicate accounts.

2) **Profile Management**
    - **Requirement:** The system should allow users to update their profile information, including contact details and professional credentials for real estate professionals.
    - **Rationale:** Ensures that users can keep their profiles up to date, which is crucial for trust and accurate communication.

3) **User Authentication**
    - **Requirement:** The system should authenticate users during login using their registered email/phone number and password, with captcha to enhance security.
    - **Rationale:** Ensures secure access to the user's account and personal information.

Property Listing Management

1) **Property Posting**
    - **Requirement:** The system shall allow users to post property listings with key details such as property type, location, price, and images.
    - **Rationale:** Facilitates the sharing of property information to attract potential buyers or renters.
    - **Note:** Future enhancements may include stricter verification of property listings for improved reliability.

2) **Property Search**
    - **Requirement:** The system shall provide advanced search functionalities, including filters for location, price range, property type, and interactive map-based search.
    - **Rationale:** Enables users to efficiently find properties that meet their specific criteria.

3) **Property Listing Management**
    - **Requirement:** The system shall offer advanced search functionalities with filters for location, price range, property type, and an interactive map-based search.
    - **Rationale:** Gives users control over their listings to keep information accurate and up to date.

Communication and Interaction

1) **Direct Messaging**
    - **Requirement:** The system shall provide a secure messaging feature allowing users and professionals to communicate directly within the platform.
    - **Rationale:** Facilitates clear and confidential communication between parties.

2) **Quotation Requests**
   - **Requirement:** The system shall allow users to request quotes from professionals for property-related services.
   - **Rationale:** Enables competitive pricing and informed decision-making for users.

Real-Time Features

1) **Interactive Map**
   - **Requirement:** The system shall feature an interactive map allowing users to view properties based on geographic location.
   - **Rationale:** Enhances the user experience by enabling visual and location-based searches.

2) **Real-Time Notifications**
   - **Requirement:** The system shall provide real-time notifications for new messages, quote responses, and updates to property listings.
   - **Rationale:** Keeps users informed of important updates and communications.

Security and Data Protection

1) **Data Encryption**
   - **Requirement:** The system shall encrypt user data, including personal information and communication, to protect against unauthorized access.
   - **Rationale:** Ensures user privacy and data security, crucial for maintaining trust.

2) **Secure Access Control**
   - **Requirement:** The system shall implement access control mechanisms to ensure that only authorized users can access specific features or data.
   - **Rationale:** Protects sensitive information and restricts access to verified users only.

Administrative Functions

1) **Account Management**
   - **Requirement:** The system shall allow admins to manage user accounts, including the ability to suspend or delete accounts that violate platform policies.
   - **Rationale:** Ensures platform integrity by enforcing community guidelines.

2) **Report Generation**
   - **Requirement:** The system shall allow admins to generate reports on user activity, property listings, and professional verification statuses.
   - **Rationale:** Provides insights for platform management and decision-making.

User Feedback and Support

1) **Feedback Mechanism**
    - **Requirement:** The system shall provide a feature for users to leave feedback or report issues regarding property listings or professional interactions.
    - **Rationale:** Allows for continuous improvement of the platform and resolution of user concerns.

2) **Help and Support**
    - **Requirement:** The system shall include a help section with FAQs and a support contact form for users requiring assistance. Users can also ask for help by sending their queries to the support team's email where the support team can get in touch with them.
    - **Rationale:** Ensures users have access to help when needed, improving user experience that can help in increasing platform growth.

5.2.NON-FUNCTIONAL REQUIREMENTS

1. **Performance:** The system should be able to perform thousands of requests from users and should be able to process up to 100 property searches per second.

2. **Scalability:** The architecture should support horizontal scaling so it can accommodate more users and data volume as the platform grows, ensuring the service remains smooth even during peak usage times.

3. **Security:** User data, especially personal information and communication must be encrypted both at rest and in transit. AES-256 and TLS 1.2 (or later) are recommended for encryption. Multi-factor authentication (MFA) should be implemented for user accounts, and API endpoints need to be secured using OAuth 2.0.

4. **Reliability:** Implement automated backups for both the database and the server.

5. **Usability:** The platform should be intuitive and user-friendly, requiring a minimal learning curve. Provide comprehensive user documentation and a help centre to assist users with common issues.

6. **Maintainability:** The codebase should follow best practices for readability, modularity, and proper documentation, adhering to coding standards. Utilize GitHub for version control and code maintenance, including tracking changes, pull requests, and issues. Implement CI/CD pipelines to automate testing and deployment, ensuring quick and reliable updates.

7. **Portability:** The system should be easily deployable on various platforms (e.g., AWS, Google Cloud, Azure, or on-premises Windows) with minimal changes. The mobile application should run seamlessly on Android devices only. The app will not be available for iOS devices.

8. **Compliance:** The platform must comply with local data protection laws in Bangladesh, ensuring user data privacy and proper handling of sensitive information. Ensure adherence to international standards for software development and data security, such as ISO/IEC 27001.

5.3.INTERFACE REQUIREMENTS

1. The platform will feature a responsive and accessible user interface (UI) for both web and mobile applications, supporting multi-language functionality in Bengali and English to cater to users in Bangladesh. The UI will feature an interactive map-based search using the Google Maps API, allowing users to explore property listings by location. The design will prioritize mobile-first accessibility, ensuring ease of use on Android smartphones, which are prevalent in the region.

2. The platform will be hosted on AWS Amplify, supporting deployment and CI/CD processes, enabling efficient updates to web and mobile applications through seamless GitHub integration. AWS Lambda will handle backend logic, providing serverless execution for key functionalities like user authentication, property management, and secure messaging. AWS API Gateway will manage communication between the frontend and backend, ensuring secure and scalable interactions.

3. Amazon RDS (MySQL) will manage user profiles, property listings, and transactional data, with AWS Lambda handling access. Media files, such as property photos, will be stored and processed in Amazon S3. The platform will use HTTPS for secure client-server interactions and WebSocket for real-time messaging, ensuring instant updates and secure data transmission.

5.3.1. SYSTEM IN CONTEXT

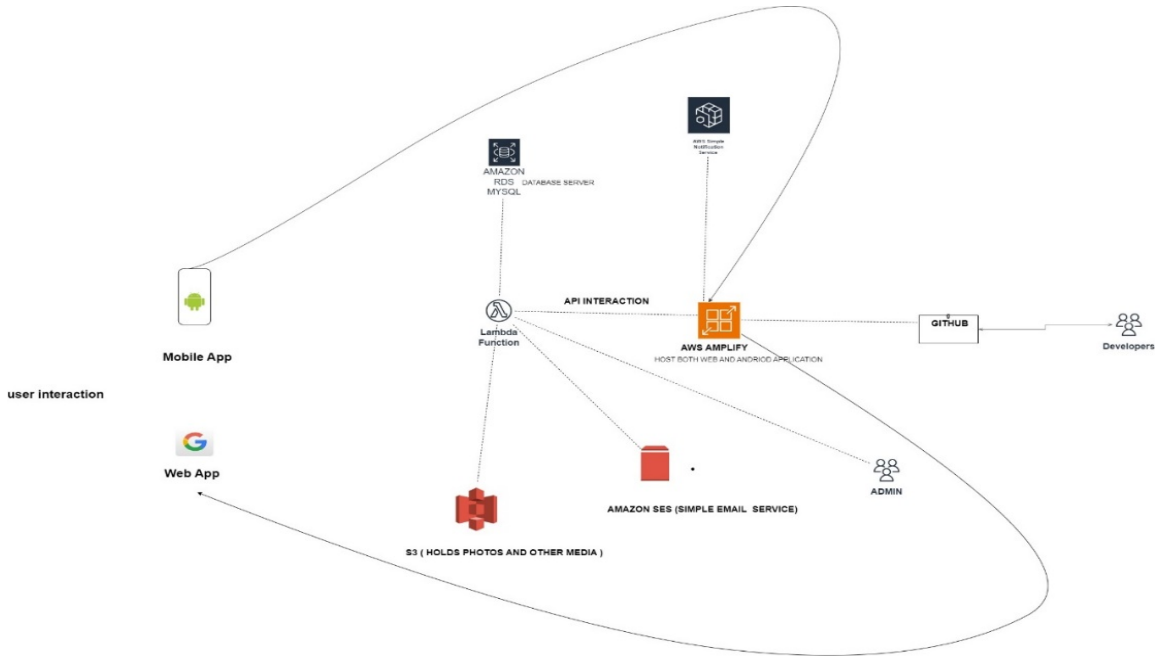

Fig2. High-Level System Architecture

The attached image presents a high-level architectural diagram for a system designed to serve both mobile and web applications. This system is likely intended for a platform that facilitates user interactions with various backend services hosted on AWS. Below is an integrated description of how the system components relate to users, developers, administrators, and other platforms, along with guidance on how to implement this architecture.

1. User Interaction:
Mobile App (Android):

- Description**:** Users engage with the platform through an Android mobile application. This app allows users to create profiles, post jobs, and interact with other users or service providers.
- Integration**:**
  - The mobile app connects to backend services via APIs exposed through AWS API Gateway and processed by AWS Lambda functions. This real-time interaction ensures users can perform tasks like profile creation, job posting, and communication with other users or admins.

- **Web App:**
  - Description**:** Users can also access the platform via a web application, providing a similar experience to the mobile app but through a web browser.
  - Integration:

- The web app is hosted on AWS Amplify, ensuring that users can access the platform from any device with a browser. Like the mobile app, it interacts with backend services through API calls, enabling real-time data processing and interaction.

2. Backend Services on AWS:

- **AWS Amplify:**
  - Purpose**:** AWS Amplify is central to hosting both web and mobile applications. It facilitates the deployment, hosting, and continuous integration/continuous deployment (CI/CD) of full-stack web and mobile applications.
  - Integration**:**
    - GitHub Integration**:** AWS Amplify is connected to a GitHub repository, enabling continuous deployment. Developers can push updates to the repository, which are automatically deployed to the live environment.
    - API Interactions**:** Amplify enables seamless API interactions between the frontend (web/mobile apps) and backend services, handling tasks like data retrieval, processing, and user authentication.

- **Amazon RDS (MySQL):**
  - Purpose**:** This managed relational database service stores all critical user data, including profiles, job postings, and transactional records.
  - Integration**:**
    - RDS communicates with the mobile and web apps through Lambda Functions, which process API calls to retrieve and store data. This ensures efficient data handling and secure storage.
    - Security**:** The RDS instance is secured using IAM roles and security groups, ensuring that only authorized entities can access sensitive data.
- **Lambda Functions:**
  - Purpose**:** AWS Lambda functions to handle the backend logic, processing API requests from the mobile and web apps, interacting with the database, and managing communication with other AWS services.
  - Integration:
    - Lambda functions serve as the intermediary between the web/mobile apps and Amazon RDS. They also interact with Amazon S3 for media handling and Amazon SES for sending emails, ensuring that all backend tasks are processed efficiently.
- **Amazon S3:**
  - Purpose: Amazon S3 stores user-generated media, such as photos and documents, which are uploaded via mobile and web applications.
  - Integration:
    - The S3 bucket is integrated with Lambda functions to process media files (e.g., resizing or compression) before storing or serving them. The mobile and web apps directly access media from S3, providing a smooth user experience for viewing and managing media files.

- **Amazon SES (Simple Email Service):**
  - Purpose**:** Amazon SES is used for sending various types of emails, including notifications, confirmations, and other user communications.
  - Integration:
    - SES works in conjunction with Lambda functions to send emails based on specific user actions or system events, ensuring timely communication with users.

- **Amazon SNS (Simple Notification Service):**
  - Purpose: Amazon SNS is used for sending push notifications to users, such as alerts, updates, or other relevant information.
  - Integration: posting, profile verification status). It can send notifications to users' mobile devices or web browsers, ensuring they stay informed in real time.

3. Development & Administration:

- **GitHub**
  - Purpose: GitHub serves as the repository for the system's source code, allowing developers to manage version control, collaborate, and continuously improve the platform.
  - Integration: Integrated with AWS Amplify, GitHub enables automated deployments whenever new code is pushed. This ensures that the system is always running the latest version with minimal manual intervention.

- **Admin Interface**

  - Purpose: The admin interface allows platform administrators to manage user accounts, verify profiles, and moderate content. This is critical for maintaining platform integrity and security.

  - Integration: Admins interact with the system via a dedicated interface that is likely hosted on AWS Amplify, with backend connectivity through API Gateway and Lambda functions. This setup provides admins with the necessary tools to oversee operations efficiently such as retrieving the emails sent from the feedback/support dialog box.

The system is designed to offer a robust and scalable platform for user interaction, leveraging AWS services to manage data, media, and communication effectively. Users can interact seamlessly via mobile and web applications, while the backend services handle all the necessary processing, storage, and communication tasks. AWS Amplify ensures that the platform is continuously updated and deployed, with GitHub integration enabling smooth collaboration among developers. Administrators have the tools to manage the platform effectively, ensuring a secure and user-friendly environment.

5.3.2. USER INTERFACE

1) **User Personas:** The team will create detailed user personas of typical Bangladeshi property listers and seekers to guide design decisions, ensuring that the interface will meet user needs and expectations.

2) **Wireframing:** Designers will create wireframes for several key screens, which include the following:
   - Home page
   - Property listing page
   - Search results page
   - User profile page
   - Messaging interface

3) **Visual Design:** Using the wireframes, the team will then create high-fidelity mock-ups that include:
   - A colour scheme that reflects Bangladeshi cultural preferences
   - Typography that supports both Bengali and English
   - Local market iconography

4) **Interaction Design:** The focus will be on developing instinctive interactions, such as:
   - Swipe gestures for viewing properties
   - Tap-to-call functionality for contacting property owners
   - Pull-to-refresh for updating listings

5) **Responsive Design:** The interface will be responsively designed to support a seamless user experience across:
   - Mobile devices [primary focus for the Bangladeshi market]
   - Tablets
   - Desktop computers

6) **Prototyping:** Interactive prototypes will be designed for testing:
   - User flow for listing a property
   - Search and filter functionality
   - Booking a property viewing

7) **User Testing:** There will be user testing sessions with local Bangladesh users to seek feedback and design iteratively.

5.3.3. HARDWARE INTERFACE

1) **Mobile Device Cameras:**

- The software will have an interface that allows users to take and upload property photos from the built-in phone camera.
- It will use standard mobile OS APIs (Android) to access camera functionality.
- The app will request camera permissions from the user upon first use of this feature.

2) **GPS and Location Services:**
   - The platform will interface with device GPS hardware for location-based services.
   - It will use geolocation APIs to access precise location data for property mapping and search functionalities.
   - The software will request location permissions, offering options for 'while using the app' and 'always' access.

3) **Network Hardware:**
   - The app will communicate with server infrastructure via cellular networks (3G/4G/5G) and Wi-Fi.
   - It will use standard HTTP/HTTPS protocols for API calls and data transfer.
   - The software will be optimized for lower bandwidth scenarios common in some parts of Bangladesh.

4) **Device Storage:**
   - The app will interface with device storage for caching data and storing user preferences.
   - It will use platform-specific storage APIs to manage local data.

5) **Display Adaption:**
   - The software will interface with various display types and sizes, adapting its UI accordingly.
   - It will use responsive design principles to adjust to different screen resolutions and aspect ratios.

6) **Audio Hardware:**
   - The app will interface with device speakers and microphones for potential voice search or audio description features.
   - It will use standard audio APIs provided by mobile operating systems.

7) **Communication Protocols:**
   - RESTful APIs over HTTPS for server communication
   - WebSocket for real-time messaging features
   - Push notification services (FCM for Android) for alerts and notifications

8) **Interface Requirements:**

- Internet connectivity (Wi-Fi or Cellular)
- GPS enabled for location-based features
- Camera access for property photo uploads
- Storage access for caching and local data management

5.3.4. SOFTWARE INTERFACE

Below are the software interfaces that will be built:-

1) The front end will be built with ReactJS, using Redux for state management and React Router for navigation. For developing a dynamic single-page application for the real estate platform, Lambda will manage API connection.

2) The backend framework will use Node.js and Express.js which will handle HTTP requests effectively. AWS MySQL will be the database type, offering the real estate platform flexible data modelling.

3) The real estate platform's storage options will be an Amazon RDS Server based on MySQL and Amazon S3. S3 will manage images and media assets, offering property visuals and scalable object storage. Structured data will be managed by the SQL-based database MySQL database, which will provide quick and flexible access to data for properties and users Together, they provide effective content distribution and strong data management throughout the whole platform.

4) Python Integration: Python 3.9 will be used for data analysis and machine learning tasks, such as property price trend analysis and recommendation systems. These Python services will be exposed as API endpoints using AWS Lambda Server, which the Node.js backend can consume as needed.

5) External APIs: AWS Lambda will be used to link external APIs to Google Maps, allowing serverless processing for image management and location services. Across the real estate platform, our method guarantees scalable and effective management of map displays and property photo operations.

5.3.5. COMMUNICATION INTERFACE

Potential communication interfaces:

1) **Local Area Network (LAN) communication**

   Purpose:

   - LAN communication enables traffic to pass through different database servers and web servers within the same network using TCP/IP for reliable flow of data.

   Use Cases:

- Developers may run the application, database, and other services on the same local network for testing and debugging.
- In a microservices architecture, different services (e.g., the web server, database server) communicate over the LAN using TCP/IP.

2) Internet communication via HTTP/HTTPS

Purpose:

The protocols facilitate communication between the client and the web server. The HTTP protocol is not secure and will be only used during the initial development phase whereas the HTTPS protocol will be used for the production phase as it adds a layer of security using encryption. HTTP protocol will be functioning on the local network at port 80 while HTTPS will be deployed on AWS or any Linux server using port 443.

Use Cases:

- API communication for CRUD (Create, Read, Update, Delete) operation for both developer and client. For example, a client browsing a rental house is a read/get operation.
- Content delivery which involves transferring items such as pictures, and webpages from the server to the client.
- Form submissions of households to advertise their house for rent.

3) FTP/SFTP

Purpose:

File transfer protocol or secure file transfer protocol is used to transfer files between the device and the server. The SFTP is secure and the FTP protocol as SFTP uses SSH (Secure Shell) for encryption and authentication. In the context of connecting developers and system servers, WINSCP can be used.

Use cases:

- Used for website development for uploading HTML, CSS and images to the web server. Installing software packages,
- Securely transferring backup files form the production server to another backup server.

CLIENT SIGN OFF

Name	Position	Signature	Date
Tanjila Kanij (Swinburne University of Technology)	Convenor	Tanjila	29/08/2024
Organisation			