**Home Rent Application Software**

**Lab-01**

**Description of Tropics**

**Tropics** is an innovative, home rent application platform specifically made for individuals and families looking for rental homes. It provides a personalized search experience with bachelors, single women, small families and larger families that match their specific housing requirements.

**Key Features:**

* **Property Search:** Allows users to filter rental properties based on criteria such as price, location, size, number of rooms, amenities, and pet-friendly options.
* **User Profiles:** Customizable profiles for bachelors, families, and other user types to ensure they find properties suited to their specific needs.
* **Document Upload:** Users can easily upload documents like IDs, salary slips, and references to expedite the application process.
* **Application Tracking:** Tenants can track the status of their rental applications and get real-time updates.
* **Communication Tools:** Messaging system that allows direct communication between tenants and landlords.
* **Payment Integration:** In-built payment options for paying deposits, rent, and maintenance fees directly through the platform.

**Stakeholder Identification**

Identifying the right stakeholders is crucial for ensuring that **Tropics** meets the needs of everyone involved in the rental process. The primary stakeholders for this software are:

* **Tenants (Bachelors, Females, Small & Big Families):**
* Primary users who will use the platform to search for homes and submit rental applications.
* **Landlords/Property Owners:**
* Post available rental properties, review tenant applications, and manage their listings.
* **Property Managers:**
* Oversee the property listing and ensure smooth communication between tenants and landlords.
* **Real Estate Agents:**
* Assist tenants in finding homes and managing negotiations between tenants and landlords.
* **System Admin:**
* Responsible for system maintenance, ensuring the security of personal data, and handling platform-related issues.
* **Service Providers (Maintenance and Legal):**
* Facilitate property maintenance and manage legal processes related to rental agreements.

**Requirement Gathering Technique**

To understand the needs of various users and stakeholders, several requirement gathering techniques were employed:

* **Interviews:**
* Detailed interviews with potential tenants (bachelors, small families, etc.) and landlords to understand their expectations.
* **Surveys:**
* Online surveys sent to a broad user base to understand preferences, such as property types, price points, and features like pet-friendliness.
* **Observation:**
* Observing users' interactions with existing rental platforms to identify pain points and areas for improvement.
* **Workshops:**
* Collaborative workshops with designers, developers, and real estate professionals to ensure all needs are captured effectively.

**Functional Requirements**

Functional requirements outline what the system must do. For **Tropics**, the key functional requirements include:

* **User Registration and Authentication:**
* Users must create accounts and log in securely to use the platform.
* **Property Search:**
* Tenants must be able to search for rental properties based on location, size, budget, family-friendly amenities, etc.
* **Profile Management:**
* Tenants and landlords can manage their profiles, including personal details and rental preferences.
* **Rental Application Submission:**
* Tenants can submit applications along with necessary documents directly through the platform.
* **Payment Integration:**
* Integrated payment gateway for secure transactions (e.g., deposit payments, monthly rent).
* **Messaging System:**
* A messaging platform for tenants and landlords to communicate directly.
* **Application Status Tracking:**
* Tenants can track the progress of their rental applications (e.g., pending, approved, rejected).
* Help Centre

**Non-Functional Requirements**

Non-functional requirements focus on how the system should perform and are equally important as functional requirements. For **Tropics**, the non-functional requirements include:

* **Usability:**
* The system must be intuitive and easy to use for all user types, ensuring a smooth user experience.
* **Performance:**
* Fast response times when users search for properties, submit applications, or make payments.
* **Scalability:**
* The system should scale to handle an increasing number of users and properties over time.
* **Security:**
* Sensitive personal information (IDs, financial details) must be encrypted and securely stored.
* **Reliability:**
* The system must be highly reliable, ensuring uptime of at least 99.9%.
* **Compatibility:**
* The platform must be compatible across various devices (desktops, tablets, and smartphones) and browsers.

**Requirement Specification**

The **requirement specification document** for **Tropics** includes:

**System Overview:**

* Provides an overall description of the platform, its objectives, and intended use cases.

**Use Cases:**

* Describes specific actions users can take, such as "As a bachelor, I want to search for single-bedroom apartments in the city centre."

**System Architecture:**

* High-level design showing the different modules and how they interact (e.g., front-end, back-end, payment systems, etc.).

**Data Flow Diagrams:**

* Illustrates how data moves between the system and its users (e.g., when a tenant submits an application or makes a payment).

**System Limitations**

Some key system constraints include:

* **Data Storage Limitations:**
* There may be limitations on the number of properties, images, or documents that can be stored in the system.
* **Legal Compliance:**
* The system must comply with local and international privacy laws (e.g., GDPR, CCPA) when handling user data.
* **Network Dependency:**
* The system requires a stable internet connection to function properly, especially for document uploads and real-time communication.

**Lab-02**

**COCOMO Model in Software Engineering**

The **COCOMO (Constructive Cost Model)** is used to estimate the time, cost, and effort required for software development.

* **Type of Model:** **Basic COCOMO Model**
* **Estimated Size of the Project:** Medium-size software application with multiple modules (property search, payment system, user management, etc.)
* **Effort Estimate:** 10-15 person-months.
* **Development Time:** Approximately 8-12 months for full development and testing.
* **Cost Estimate:** Between $150,000 and $250,000 depending on the development team size, technologies, and integrations.

**Lab-03**

**Use Case Diagram**

Here’s a **Use Case Diagram** illustrating the primary interactions in **Tropics**:

**Tenant (Bachelor, Female, Small/Big Family):**

* Search for homes
* Submit applications
* Communicate with landlords
* Track application status

**Landlord/Property Owner:**

* List properties
* View applications
* Communicate with tenants

**System Admin:**

* Manage users
* Monitor system performance
* Ensure data security

**Lab-4**

**Prototyping**

An **interactive prototype** of **Tropics** was created using tools like **Figma** and **InVision** to visualize the user interface and the flow of the system.

* The prototype allowed stakeholders to interact with the application before full-scale development began, ensuring that the design met user needs.
* Feedback gathered from users helped improve usability, especially for those with specific housing requirements (e.g., bachelors or large families).

**Lab-05**

**Testing**

Testing is a critical phase to ensure the software works correctly and meets user expectations. For **Tropics**, the following types of testing were performed:

**Unit Testing:** Testing individual components of the system (e.g., search function, application submission).

**White Box Testing : Testing** is a software testing technique where testers have access to the internal code and design of the software. It focuses on checking the program's logic, flow, and functionality.

**Key Techniques:**

* **Code Coverage**: Ensuring all code lines, branches, loops, and paths are tested.
* **Path Testing**: Verifying all possible execution paths are covered.
* **Loop Testing**: Testing loops for boundary conditions.
* **Control Flow Testing**: Analyzing how control flows through the program.

**Black Box Testing** is a method where testers check the software's functionality without knowing how it works internally. They focus on inputs and expected outputs, based on the software’s requirements.

### Key Points:

* **No Code Knowledge**: Testers don’t need to understand the code.
* **Focus on Functionality**: Ensures the software works as expected.
* **External Testing**: Testers provide inputs and check the outputs.

### Types of Black Box Testing:

* **Functional Testing**: Ensures the software meets its requirements.
* **Non-Functional Testing**: Tests performance, usability, and security.
* **Regression Testing**: Checks if new changes affect existing features.
* **Acceptance Testing**: Verifies if the software meets user needs and is ready for use.

**Security Testing:** Ensuring that all sensitive information is encrypted and properly protected.

**Conclusion**

In conclusion, **Tropics** is a powerful and user-centric platform designed to simplify the home rental process for bachelors, families, and other users. By incorporating robust features, a personalized experience, and a secure environment, Tropics aims to revolutionize how tenants and landlords connect, communicate, and complete rental agreements.