1. PROJECT PROPOSAL

1.1 Background to the Problem

In many urban and semi-urban areas, managing rental properties is still handled through traditional and manual processes. Landlords, tenants, owners, and managers often depend on phone calls, handwritten records, spreadsheets, or verbal agreements to manage rental activities. These methods are inefficient, error-prone, and lack transparency. As a result, common problems arise such as unclear rental agreements, delayed rent payments, poor maintenance tracking, loss of important records, and miscommunication between involved parties.

From a tenant's perspective, finding a suitable rental property, tracking application status, paying rent, and requesting maintenance services can be time-consuming and inconvenient. Tenants often do not have a clear view of rent due dates, payment history, or the progress of maintenance requests. This leads to dissatisfaction and trust issues.

Landlords and property owners also face challenges in managing multiple properties and tenants simultaneously. Keeping track of rent collection, tenant applications, maintenance issues, notices, and financial summaries manually becomes difficult as the number of properties increases. Managers who are responsible for overseeing operations lack a centralized system to monitor activities, assign maintenance tasks, and generate reports efficiently.

To overcome these challenges, a Home Rental Management System is required. This system aims to provide a centralized, web-based platform where tenants, landlords, owners, and managers can interact through a secure and transparent environment. By automating key rental processes such as property listing, tenant application, rent payment, maintenance management, and notices, the system reduces manual workload, minimizes errors, improves communication, and ensures better control and efficiency in rental management operations.

1.2 Selection of Process Model

For the Home Rental Management System, the Scrum (Agile) process model has been selected as the most appropriate software development approach. This system involves multiple user roles—Tenant, Landlord, Owner, and Manager—each with different responsibilities and system interactions. The presence of diverse roles and interconnected modules makes the development process dynamic, where requirements may evolve during implementation. Scrum is well-suited to handle such complexity due to its flexible and iterative nature.

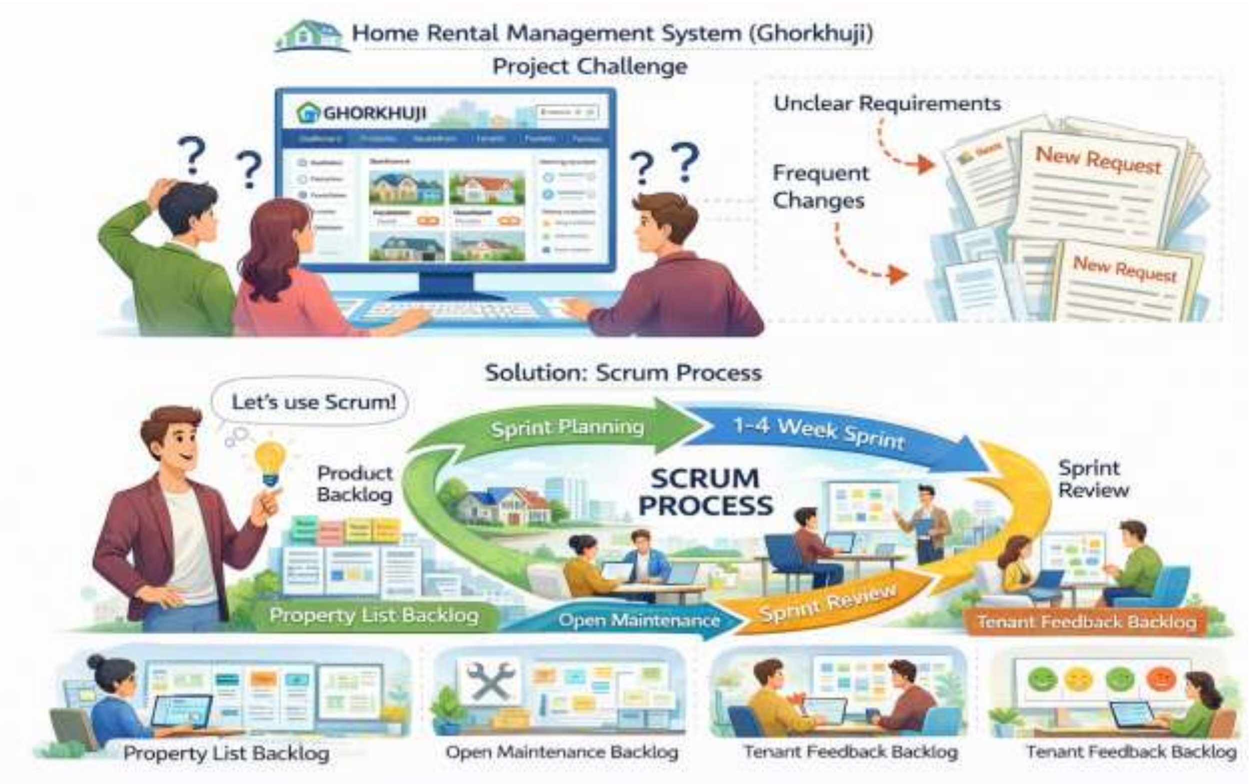
The Home Rental Management System includes several functional components such as user authentication, role-based dashboards, property listing and approval, tenant application handling, rent payment processing, notice distribution, and maintenance management. Developing all these features in a single phase would be difficult and risky. Scrum addresses this challenge by dividing the development process into short, structured iterations called sprints, allowing the team to focus on a limited set of features at a time.

In Scrum, all system requirements are maintained in a product backlog, which is continuously refined and prioritized. Before each sprint, the team selects a group of high-priority features during sprint planning and commits to delivering them within a fixed time frame. Daily scrum meetings help team members share progress, identify obstacles, and maintain alignment. At the end of each sprint, a functional version of the system is reviewed and evaluated, followed by a retrospective to improve the development process.

Illustrative Sprint Plan

- **Sprint 1:** User registration, login, and role-based access control
- **Sprint 2:** Property listing, tenant application submission, and approval workflow
- **Sprint 3:** Rent payment system, notices, and maintenance request management

This gradual development approach ensures that each module is properly designed, implemented, and tested before moving to the next stage.



Rationale for Choosing Scrum

One of the primary reasons for selecting Scrum is its ability to accommodate changing requirements. In a rental management system, business rules such as rent policies, maintenance procedures, or approval flows may need adjustments as development progresses. Scrum allows such changes to be introduced without restarting or restructuring the entire project.

Scrum also emphasizes early delivery of working software. By producing a usable system increment at the end of each sprint, the team can validate functionality early, detect design flaws, and reduce the risk of major failures near project completion. This is particularly valuable in an academic project where time is limited.

Another important advantage of Scrum is its focus on continuous feedback and quality improvement. Sprint reviews enable stakeholders and team members to evaluate completed features and suggest refinements. Continuous testing during each sprint helps identify defects early, improving system reliability and maintainability.

From a teamwork perspective, Scrum is highly effective for group projects. It promotes collaboration, clear task ownership, accountability, and regular communication among team members. These characteristics make Scrum especially suitable for student-based projects where coordination and progress tracking are essential.

Why Scrum Is Suitable for This Project

Scrum was chosen because it:

- Supports incremental and modular system development
- Adapts well to evolving requirements
- Encourages regular communication and teamwork
- Delivers functional system components early
- Improves software quality through continuous testing and review

Why Other Process Models Were Not Selected

Process Model	Reason for Exclusion
Waterfall Model	Its rigid structure does not allow changes once development begins, which is unsuitable for evolving system requirements.
V-Model	Although testing-focused, it follows a sequential approach that limits flexibility and change management.
Extreme Programming (XP)	Requires continuous customer involvement and advanced development practices, which are difficult to ensure in an academic group project.
Prototyping Model	Focuses mainly on user interface development and may result in incomplete system architecture if used alone.
Iterative Model	Lacks defined roles, sprint planning, and structured feedback compared to Scrum.
DSDM	Involves heavy documentation and strict control mechanisms, making it impractical for student projects.