Android Application for Employee Rewarding and Management System using Kotlin [ERMS]

**Mid Evaluation Report**

By

Sabghat Ullah Khan (CS120192060, [sabghat90@gmail.com](mailto:sabghat90@gmail.com))

Saleh Hayat (CS120192061, [salukhattak2000@gmail.com](mailto:salukhattak2000@gmail.com))

**Supervisor**

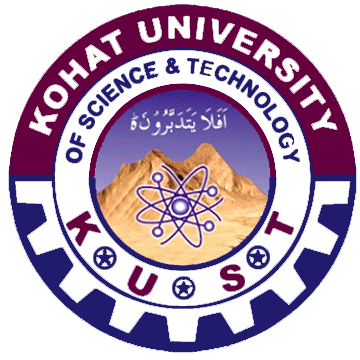
Mr. Qadeem Khan

Institute of Computing

KUST, Kohat

\_\_\_\_\_\_\_\_\_\_\_

Signature



**Institute of Computing**

**Kohat University of Science and Technology, Kohat-26000**

**Khyber Pakhtunkhwa, Pakistan**

**Session (2019 – 2023)**

*Chapter 02*

***Requirement Analysis***

1. **Project Requirements Gathering Process:**

The requirements which we gatherd from multiple sources are the following:

* Visited different sites across web to for requirements gathering.
* Conducted face-to-face meetings with IBS students.

As a part of analysis different websites are visited and read different research papers.

* <https://www.solvexia.com/blog/what-is-office-automation-your-complete-guide>
* <https://www.shrm.org/resourcesandtools/tools-and-samples/toolkits/pages/understanding-organizational-structures.aspx>
* <https://www.managementstudyguide.com/motivation-and-financial-and-non-financial-rewards.htm>
* <https://factohr.com/policy/employee-recognition-and-reward/>

1. **Feasibility Study:**

A feasibility study is an evaluation of a proposed project or system to determine whether it is technically and economically feasible, and whether it is the best course of action. The purpose of a feasibility study is to analyze and determine the potential success of a proposed project before investing time and resources into its development.

The study typically includes the following steps:

1. **Problem Definition:**

The problem definition for the Employee Rewarding and Management System project is the need for a more efficient and effective way to manage employee attendance, work, salary slips, events, and performance in an organization. Currently, many organizations face challenges in managing these tasks manually, which can lead to errors and inefficiencies. Additionally, employee rewards and recognition programs are important for improving employee satisfaction and motivation, but often lack proper tracking and management systems. The proposed solution aims to address these problems by providing a centralized and automated platform for employee management and rewarding, enabling organizations to improve their efficiency, effectiveness, and employee engagement.

1. **Technical Analysis:**

For the technical analysis of the Employee Rewarding and Management System (ERMS) project, the following aspects should be considered:

* **Platform Compatibility:** The ERMS application is planned to run on Android Nugget 7.0 and above versions.
* **Technical Feasibility:** The technical feasibility of the project should be evaluated in terms of the availability of resources and skills needed to develop the application.
* **Integration with existing systems:** If the ERMS application needs to be integrated with any existing systems or tools, the feasibility of this integration should be analyzed.
* **Security:** The security of the ERMS application should be evaluated to ensure that sensitive employee information is protected and secure.
* **Performance:** The performance of the ERMS application should be evaluated to ensure that it can handle the expected workload and provide a fast and smooth experience for users.
* **Scalability:** The scalability of the ERMS application should be analyzed to ensure that it can accommodate future growth and changes in the organization.
* **Maintenance:** The maintenance and support needs of the ERMS application should be evaluated to ensure that it can be sustained over the long-term.

1. **Risk Analysis:**

Risk analysis is an important aspect of project management, and it involves identifying and assessing potential risks that may impact the successful completion of the project. For the Employee Rewarding and Management System (ERMS) project, some potential risks might include:

* **Technical Risks:** This project involves the development of an Android application, and there could be risks associated with software development such as bugs, compatibility issues, and security vulnerabilities.
* **Data Security:** ERMS will be handling sensitive employee information, and there is a risk of data breach or theft. Measures need to be taken to secure the data and prevent unauthorized access.
* **User Adoption:** The success of the ERMS application depends on the adoption of the application by employees and managers. If the users are not comfortable using the application, it might lead to a low level of usage, and the project may not meet its objectives.
* **Time and Cost Overruns:** The development and implementation of the ERMS application can be time-consuming and expensive, and there is a risk of exceeding the estimated time and budget.
* **Maintenance and Upgrades:** The ERMS application will require regular maintenance and upgrades to stay current with changes in technology and user needs. There is a risk that the application will become outdated and may not meet user needs over time.

1. **Process Model:**

The Waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards through the phases of Requirements gathering and analysis, Design, Implementation (coding), Testing, Deployment and Maintenance.

In this model, each phase of the project must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is best suited for projects where the requirements are well understood, and there is a low risk of changing requirements throughout the development life cycle.

In the context of the Employee Rewarding and Management System (ERMS), the Waterfall model could be applied in the following manner:

* **Requirements gathering and analysis:** This phase involves understanding the functional and non-functional requirements of the ERMS application and documenting them.
* **Design:** In this phase, the architecture of the ERMS application is defined, the database structure is determined and a detailed design of the user interface is created.
* **Implementation:** This phase involves writing the code to implement the design and develop the ERMS application.
* **Testing:** This phase involves testing the ERMS application to ensure that it meets the requirements and functions as intended.
* **Deployment:** In this phase, the ERMS application is deployed in the production environment and made available to the end-users.
* **Maintenance:** This phase involves fixing any bugs, adding new features and making changes to the ERMS application as required.

1. **Functional Requirements:**

Functional requirements are the specific functions and features that a system must have to fulfill the needs of its users. In the case of the Employee Rewarding and Management System (ERMS) Android application, the functional requirements may include:

* **Employee Management:** The ability to add, update, and delete employee information.
* **Attendance Management:** The ability to mark employee attendance, including half-day and overtime options.
* **Salary Slip Generation:** The ability to generate salary slips for employees and deliver them through various channels such as email, WhatsApp, or contact number.
* **Summary Report Generation:** The ability to generate summary reports for employees.
* **Event Management:** The ability to create events and send push notifications to selected employees.
* **Meeting Management:** The ability to create and send broadcast invitations for meetings.
* **Task Management:** The ability to set tasks for employees and send notifications to the concerned employee.
* **Request Approval:** The ability for the manager to approve requests from employees such as leave requests, task completion requests, and salary slip requests.
* **Announcements:** The ability for the manager to make announcements such as employee promotions, birthdays, holidays, etc.

1. **Non-Functional Requirements:**

Non-functional requirements are the constraints and characteristics that define the quality and performance of a system. Some of the non-functional requirements for the ERMS Android application may include:

* **User-friendly interface:** The application should have an easy-to-use and intuitive interface.
* **Performance:** The application should be fast and responsive, with minimal lag time.
* **Security:** The application should have strong security measures to protect employee information and data.
* **Scalability:** The application should be scalable to accommodate an increasing number of users and employees.
* **Compatibility:** The application should be compatible with a range of Android devices and operating systems.
* **Reliability:** The application should be reliable and available for use at all times.
* **Maintenance:** The application should have a well-defined maintenance process to ensure ongoing stability and performance.

1. **Module Wise Requirements Analysis:**

Module-wise requirement analysis is a process of breaking down the whole project into smaller and manageable modules, and then analyzing and defining the requirements for each module in detail. It helps to get a clear understanding of the functional and non-functional requirements of each module, which can then be used to guide the development and testing of that module.

In the case of the Employee Rewarding and Management System (ERMS) project, the following modules can be identified:

* **Admin/Company Module:** The company module will serve as the central hub for all employee-related operations within the ERMS system. Its main functions will include:
  + **Adding employees:** The module will allow the admin/manager to add new employees to the system by providing their basic details such as name, contact information, job title, and any other relevant information.
  + **Managing employees:** The company module will provide options for updating or deleting employee information in the ERMS system. This will ensure that the data is up-to-date and accurate at all times.
  + **Selecting managers:** The module will also allow the admin to select managers from the employee list, and assign them with additional responsibilities such as conducting meetings, setting tasks, and approving requests.
  + **Employee data security:** The company module will also ensure that all employee data is securely stored and protected, and only authorized personnel have access to it.

In summary, the company module is an essential component of the ERMS system that provides a centralized platform for managing employee information and conducting related activities.

* **Admin/Manager Module:** This module will provide all the functionalities required by the admin or manager to manage the employees. This includes adding employees, updating or deleting employee records, marking attendance, generating salary slips, creating events, conducting meetings, setting tasks, and approving requests.
* **User/Employee Module:** This module will provide all the functionalities required by the employees to manage their own performance and requests. This includes viewing their performance, marking tasks complete, requesting for leave, salary slips, and loans, viewing other employees' attendance, submitting complaints, and more.
* **Rewarding Module:** This module will track the attendance, tasks, events, meetings, and work of the employees, and assign points to each employee. Based on these points, the organization will select an employee for reward on a monthly, quarterly, half-yearly, or yearly basis.

Chapter 03

***Design And Implementation***

1. **Use Case Diagram:**
2. **Company:**



1. **Manager:**



1. **Employee:**



1. **Data Flow Diagram:**
2. **Level 0:**



1. **Level 1:**



1. **Activity Diagram:**
2. **Company:**



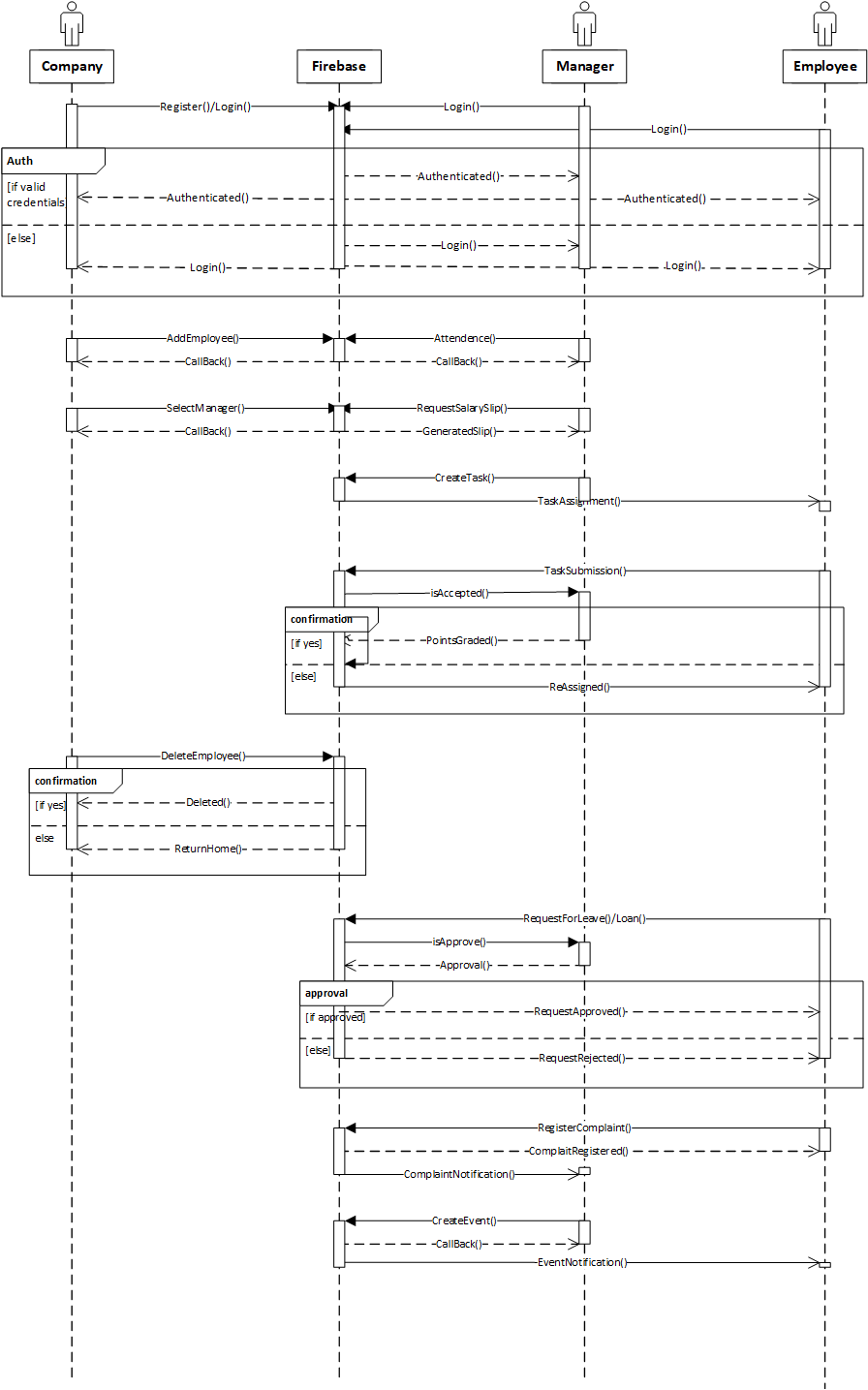
1. **Manager:**



1. **Employee:**



1. **Sequance Diagram:**

****

1. **ER Diagram:**



1. **Architecture Diagram:**



Chapter 04

***Tools and Technologies***

The tools and technologies we are using are:

* Kotlin
  + View Model
  + Live Data
* MVVM Architecture
* Android Studio
* Hilt Dependency Injection
* Firebase
  + Firebase database
  + Firebase firestore
  + Firebase auth
  + Firebase massages
* XML
* External Dependencies
* Third Party plugins

Experience level of these technologies:

**Sabghat Ullah Khan**

* Kotlin (Average)
* MVVM architecture (Average)
* Hilt DI (Average)
* Firebase (Basic)

**Saleh Hayat**

* XML (Average)
* Libraries use

# REFERENCES

1. Project GitHub Repository: <https://github.com/sabghat90/ERMS>