



Addis Ababa University
Addis Ababa Institute of Technology



ADDIS ABABA UNIVERSITY (AAU)

ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)

INFORMATION TECHNOLOGY AND SCIENTIFIC COMPUTING (ITSC)

SOFTWARE ENGINEERING

DESIGN PATTERN PROJECT ONE

SALARY MANAGEMENT SYSTEM

GROUP MEMBERS

NAME	ID NUMBER
1. KENA WAKWOYA	ATR/4296/08
2. ETSEGENET MELESE	ATR/8845/08
3. MILKIAS TONJI	ATR/8137/08

SUBMITTED TO MR. ALAZAR ALEMAYEHU

Problem Definition

In this project we are going to implement salary management system for employees in different companies and organization. In our country there are many employees that are working in the many organizations and institutions but unfortunately there is no salary management systems which manages and manipulate the salaries of employees. Because of this, many employees are facing the problem of having their salaries properly, to get rid of this problem, we have decided to implement simple salary management system that can be applied in any organizations and companies. Every detail of our application is explained with respect to the features. We are favored to implement the problem with android because, every employee and employer are almost supposed to have (the device is available) the device which they can run easily. The last but not the least, our app is very easy to manage the salary of the employee depending on the rules (that the organizations or companies have). If the employee fails to accomplish the rules and regulations, he/she will get their salary deducted. So, the admin takes attendance every day so that the consistency of salary could be easy to be managed.

The Salary Management System deals with the financial aspects of employee's salary, allowances, deductions, gross pay, net pay etc. and generation of pay-slips for a specific period. The outstanding benefit of Salary Management System is its easy implementation. Employee Salary management system helps you easily manage monthly salary & define slabs for CTC. The data is the most important thing in any organization and so it must be protected by malicious intended users. Besides these, Salary management also allows users to print the salary slip for a particular employee.

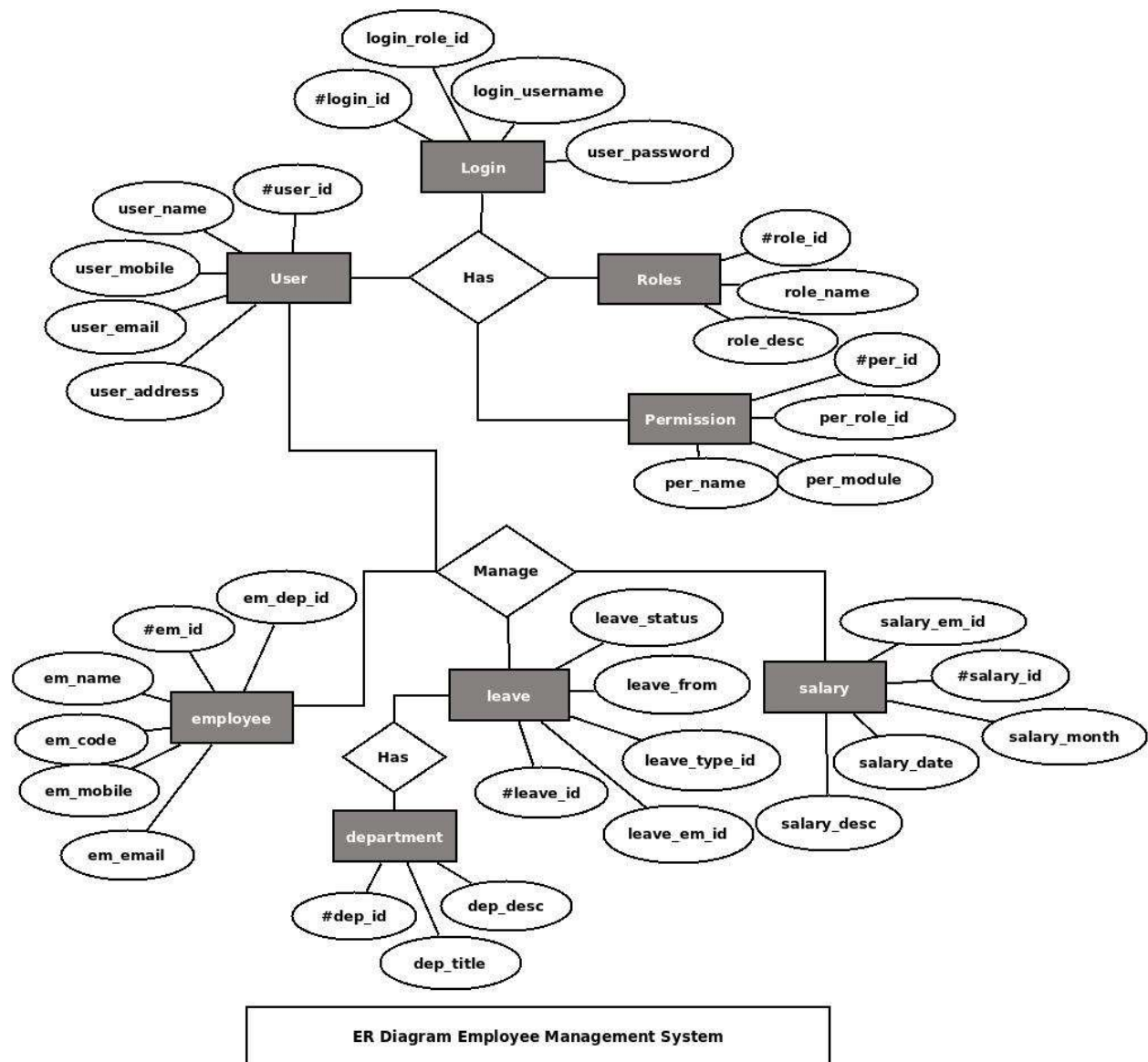
Features of salary management application

- ✓ Register employee
- ✓ Search employee
- ✓ Salary deduction
- ✓ Payment check
- ✓ Allowance

Modules of Salary Management System

- Login Module
- Employee Module
- Payment module
- Update module

ER diagram for employee management system with their salary inclusive



In salary management system there are two main actors or sometimes called users. Namely admin and employer and their role is shown below respectively.

Admin: is the one who controls the system (salary management system). He/she is capable of doing the following

- Manage employees
- Add news
- Manage news
- Take attendance
- Manage salary
- View list of the employees
- View profiles
- Change password
- Provide rules and regulations
- View complains

Employer: is the one who can get access of application by performing the following operations

- View profiles
- View rule and regulation of the company
- Can complain to the actions
- View news and meetings
- Participate and collaborate on some issues
- Register
- Login
- Change password

As it is tried to elaborate the functions of main actors of the system above, it is going to be implemented in such way.

Role of Design patters and design patterns involved in the application

- As it is required in the specification of project we can't say hundred percent implemented it but we have tried our best to match involve about 4 design pattern

solutions that supposed to help us to implement the problem. So, here are the 4 design pattern solutions that we have used

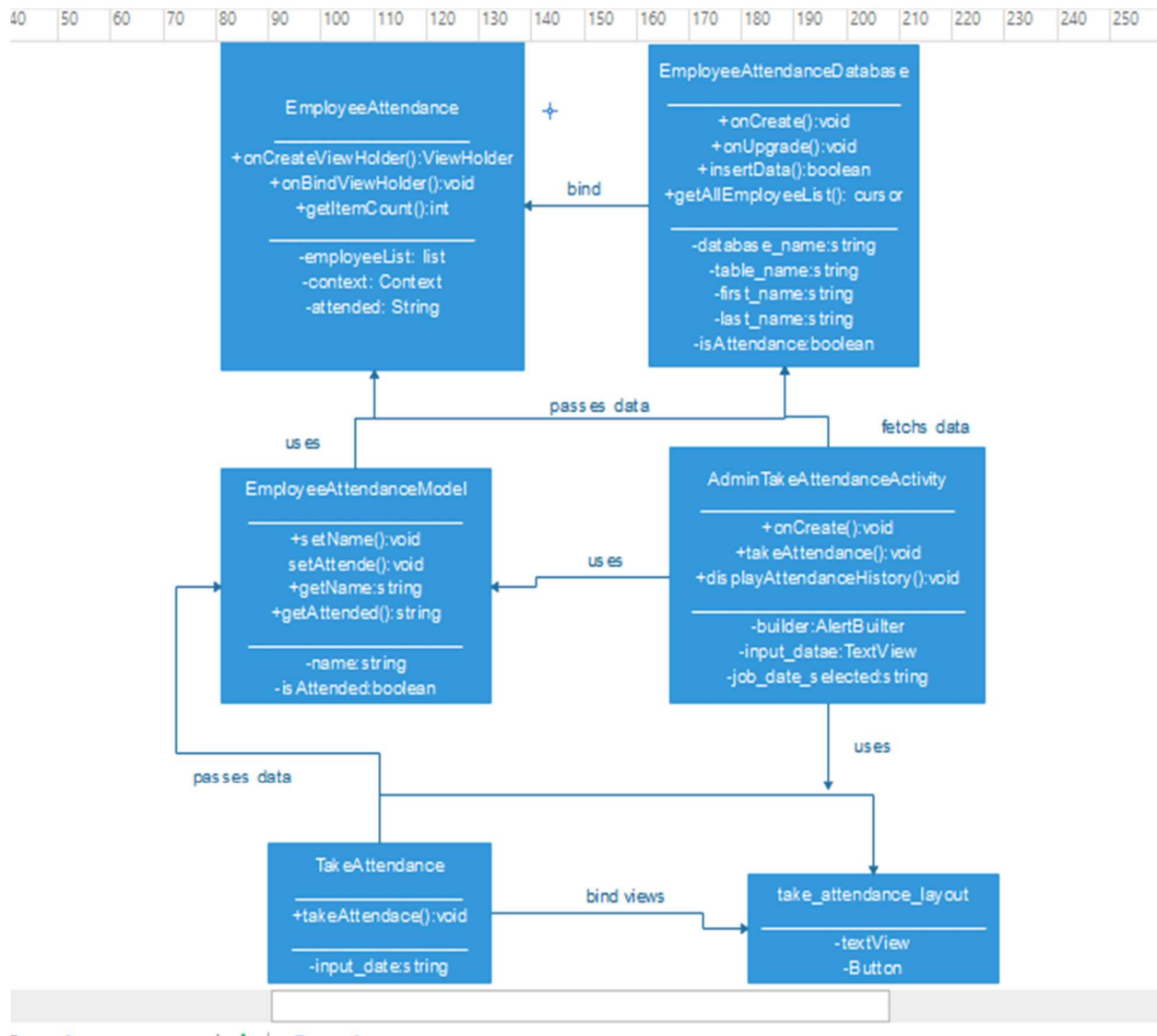
Singleton pattern

- This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.
- So, as its definition illustrates we have used this design pattern solution to instantiate database classes
- In sqlite database we extended the class from default openHelper class so that we get a single instance of the database object

Builder pattern

- Builder pattern simplifies object creation in very clean and readable way. It's very helpful when we have some model classes with many parameters. We can make some of them optional or required, and we don't force the user to use specific order (as in the constructor). By using Builder pattern we result with the elegant chain of the methods. The most common use is in `AlertDialog.Builder()` class:
- Builder pattern is a great approach, not only for model classes but for every object which has more than three or four parameters. With a bit of additional work, we can increase the readability of our code. Design patterns are recognized as the best practice, so it's big advantage if you know some of them and Builder is a good one to start with.
- We have done plenty of things by using builder class like using default builder class to create many dialog boxes
- When we take input of the date from user we used alert dialog builder that made everything easy for us.

Class diagram for builder design pattern



Model View Controller pattern

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

- **Model** - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
- **View** - View represents the visualization of the data that model contains.

- **Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

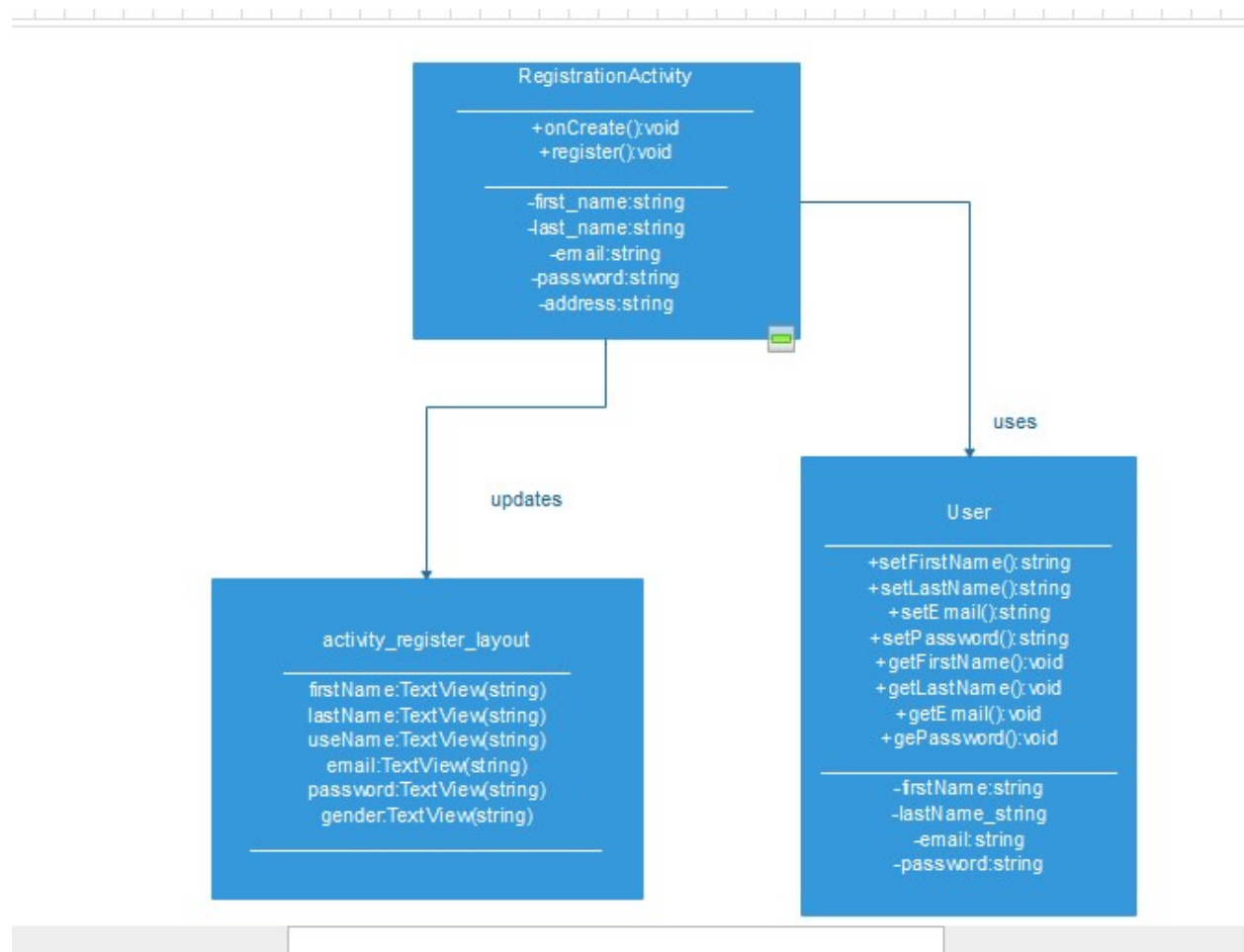
Almost all part of the salary management system is built using this design pattern solution. Our application has three phases like adapter, models and activity.

Each part of the application has different functionality like adapter is used to bind views from layout file and passing it to the class behaving controller in our case activity.

Model classes that we have used to store data:

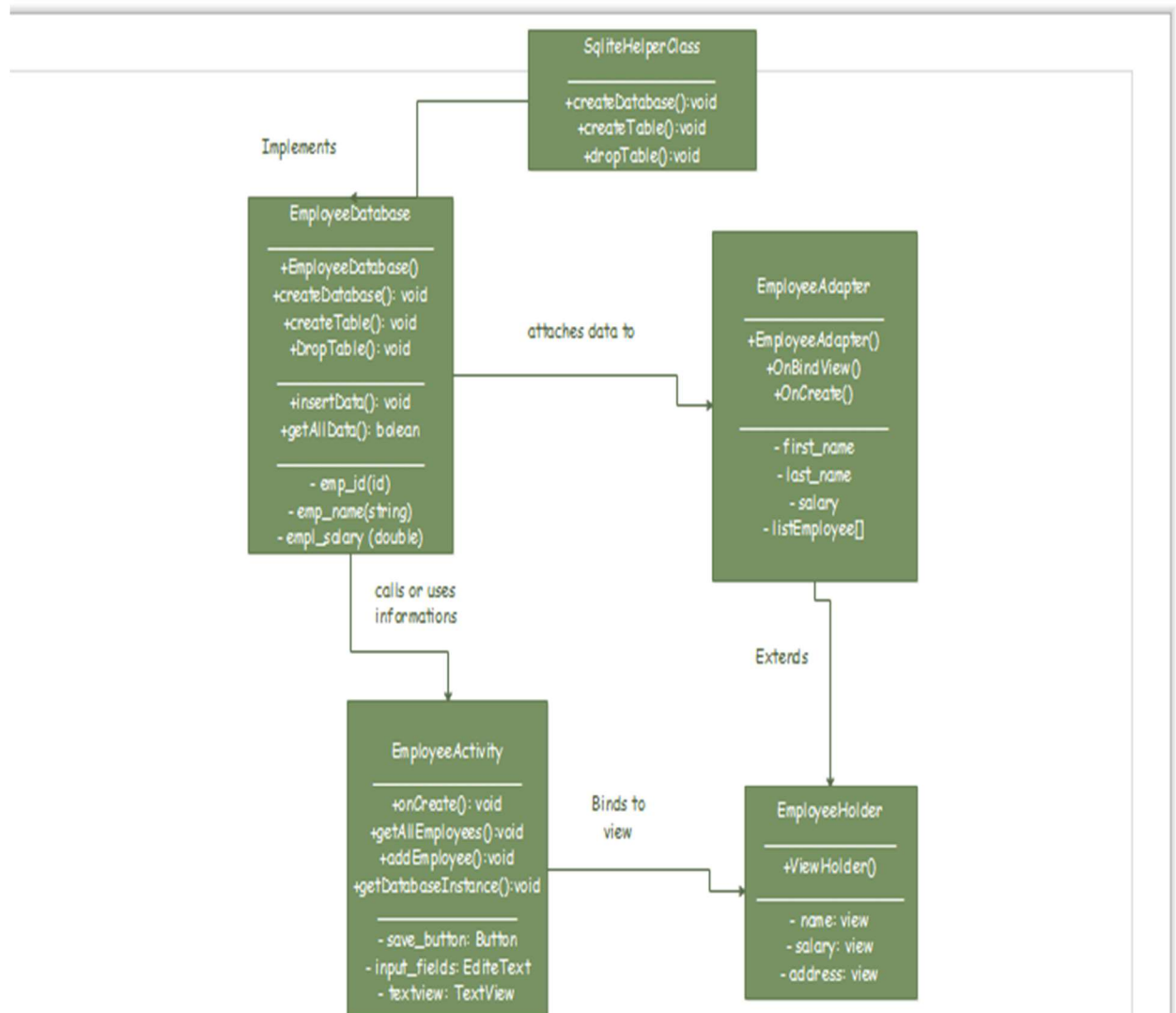
- Employer list
- Users for registration purpose
- News Model class
- EmployerAttendance class
- userRegistration database class
- AttendanceHistoryModel class
- And others

Class diagram for MVC design pattern



Class diagrams

We have created some demo of our class diagram of single class called EmployerActivity including its functionality. Adding all the class diagram to the document is like duplication of the same diagrams with a little bit change, so instead of doing this, we have shown the only single class diagram of selected class.



CONCLUSION

In this document it is tried to explain the detail of existing problem and provided somehow likely solutions that we thought could solve the problem. Every detail of implementation is also included as much as possible. This document is not the best to express whole idea of the project because the idea is very big and can be expanded more. Additionally, all the details are not included since we included only the main idea and didn't want to make page number too many that might be not understandable. Class diagrams are the main part of the implementation and we added only one for illustration purpose.