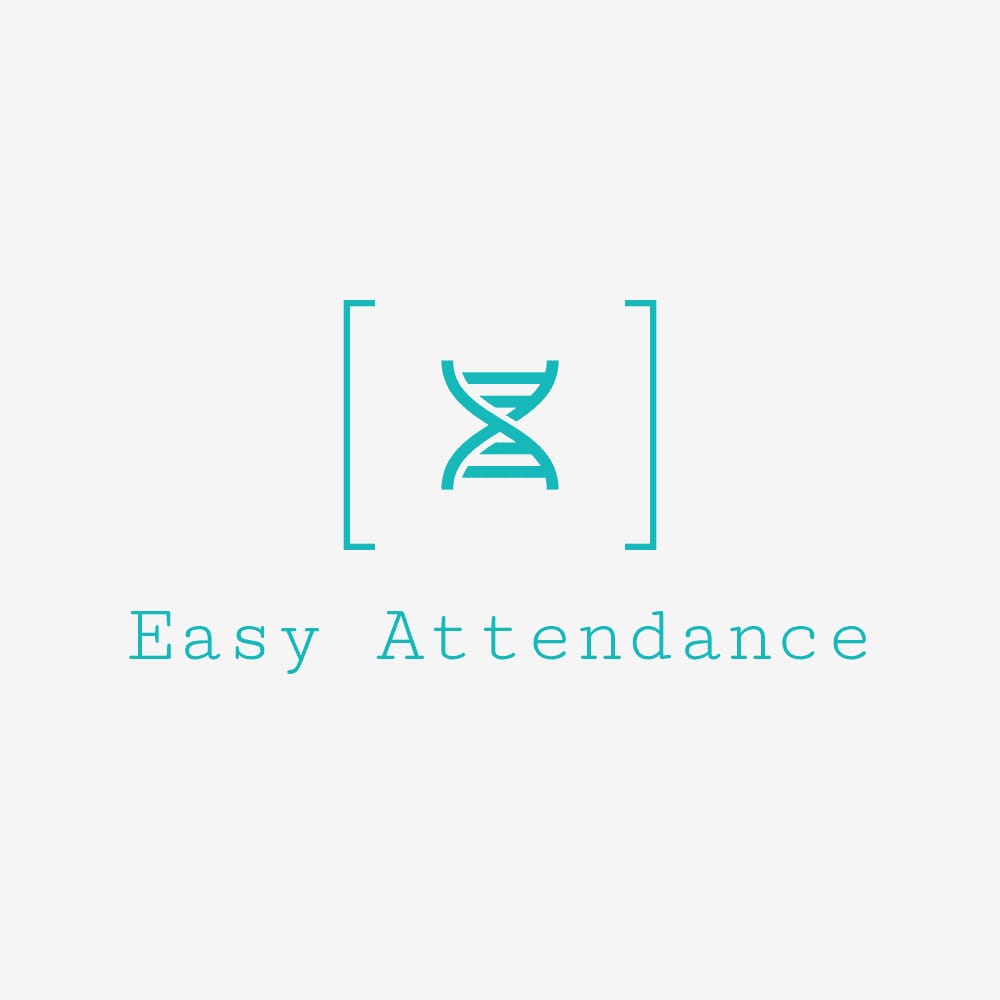
Easy Attendance

SDD



Prepared by

Danielle Zand 313290363

Shahar Furer   
Tehila Uzan  
Ester Abargel

Date: 02/02/2021

**TABLE OF CONTENTS**

**1. INTRODUCTION 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

**2. SYSTEM OVERVIEW 4**

**3. SYSTEM ARCHITECTURE 5**

3.1 Architectural Design 5

3.2 Decomposition Description 5

3.3 Design Rationale 6

**4. DATA DESIGN 7**

4.1 Data Description 7

4.2 Data Dictionary 7

**5. COMPONENT DESIGN 8**

**6. HUMAN INTERFACE DESIGN 9**

6.1 Overview of User Interface 9

6.2 Screen Images 9

6.3 Screen Objects and Actions 10

**7. REQUIREMENTS MATRIX 11**

1. **INTRODUCTION**
   1. **Purpose**

This software design document describes the architecture and system design of solving the problem of managing employees attendance out of office and the precise implementation and details required to satisfy the requirements as specified in the Software Requirements Specifications (SRS). It is assumed that the reader read the SRS, since this document also defines the implementation details of the design behavior given the requirements within it.

* 1. **Scope**

Background - Within high tech companies, and many other industries, there is an accessibility problem in maintenance of the attendance reports of the employees. The problem is that the main functionality could be preformed mostly from the office, and because of that a lot of employees that works outside from office, or from home, could not report daily on their working hours. More than that, they need their work computer to get their hours report and so on.

A desirable purpose of our system is the ability to easy maintain attendance details using phone application and without extra effort.   
In addition to the main functionality of daily reports, the system will provide extra unique feature such as forget to report notification, salary calculator, manger extra functionalities as modify workers salary, etc.

* 1. **Overview**

The design description defined in this document serves multiple purposes:

* To describe the functional structure, data, and algorithms to be implemented.
* To identify required system resources.
* To be used to assess the impact of requirement changes.
* To be used to verify compliance with requirements.
* To aid in maintenance activity.

1. **System Overview**

Our digital attendance clock will provides a daily and consistent report, about the attendance of employees in the organization.

Which allow the employee and the employer to know, that there is a neutral device that counts the working hours from the moment the employee arrives at the organization until he leaves without the working hours being edited and changed.

The digital clock will enable two fast option to clocking-in and clocking-out.

One way is Manually- quickly and conveniently with the help of a user-friendly app.

Another option is automatic- with the help of track location. The application will detect when the employee enters or leaves his workplace and will automatically update the time of entry / exit.

In addition, the system will alerts to the user when his salary change or when he forget to set his hours etc.

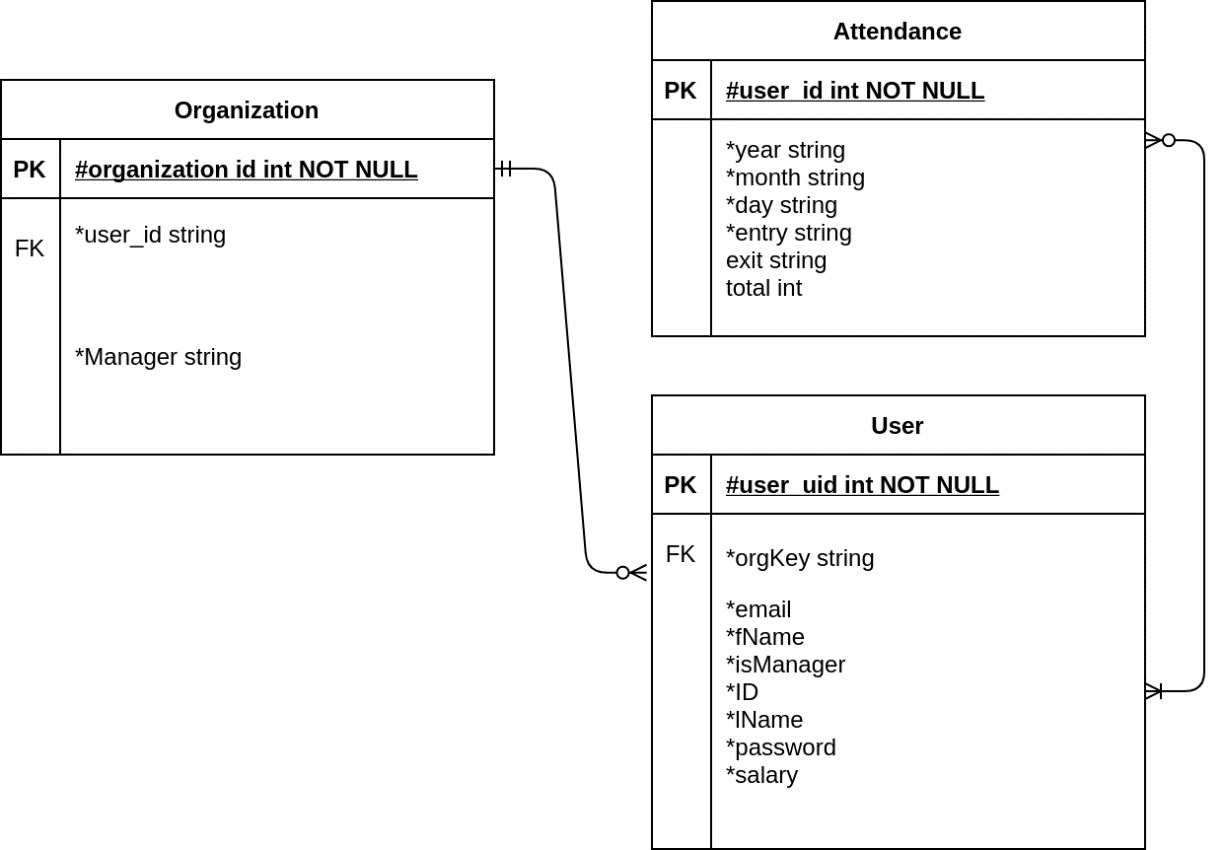
The users could easily calculate his salary by choosing the desired month.

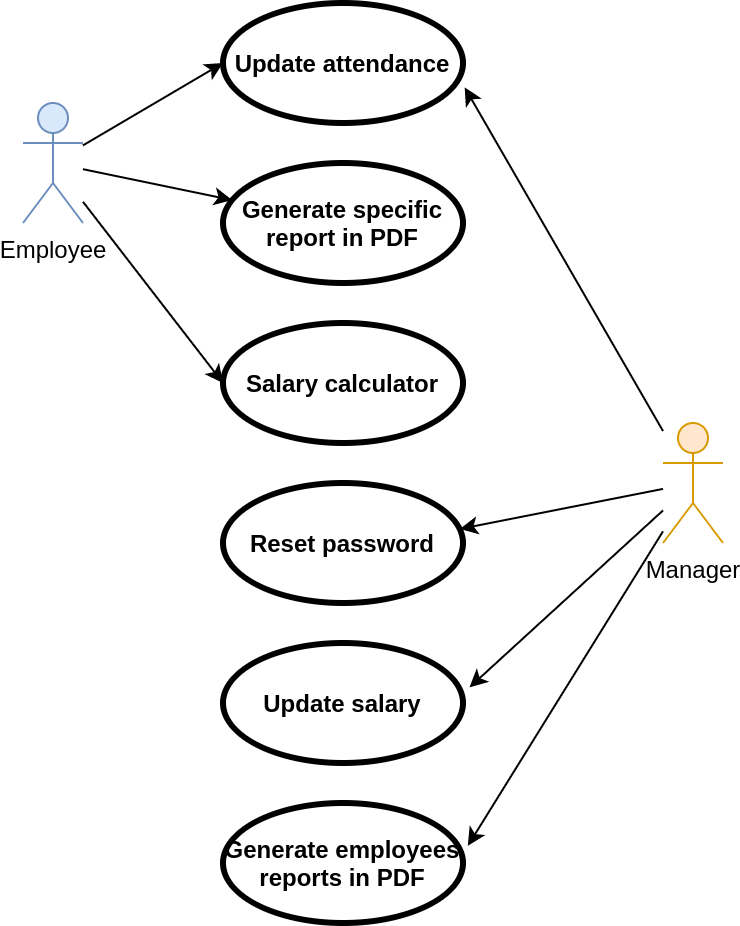
Both the employee and the employer could easily export/open monthly attendance reports from the app.

The system will have two different types of users with different authorizations:

Administrator user – In addition to all the functionality that the regular employee have, this user will have a full access to export the attendance reports of all his employees and to set their salary.

The sub user is the employee itself, that can clicked-in and clicked-out his works hors, can calculate is salary and export his monthly reports.

1. **System Architecture** 
   1. **Architectural Design**
   2. **Decomposition Description**



**3.3 Design Rationale**

We choose to use the architecture described in part 3.1, in order to create an separate component for each subsystem. This way each subsystem will have its own functionally. The advantages of using this type of architecture are- reducing the resources cost, improving the system modularity and makes it more flexible for changes.

1. **Data Design**
   1. **Data Description**

The data will be transformed into class object entities and will be stored in a database. Which will help to analysis, to do efficient search and to Query the DB. the database will store information on the employees, users with permissions, employee's attendance and list of organizations.

* 1. **Data Dictionary**

Entities list:

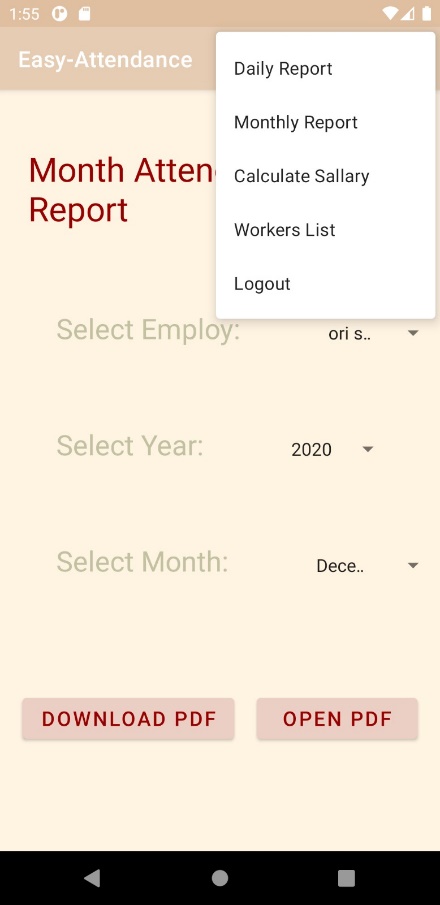
* Users – show list of employees and his Identifying details.
* Organization – show list of all the organization that have been integrated into the system. Each organization have a list of his employees
* Attendance – show list of employees and his date and time of the entry and exit.
* Massages – show list of employees and his massage.
* Locations – show list of employees and his work location.

1. **Component Design**

* New user
  + Register new users
  + Grant permissions to manager
  + manage current users.
* Daily attendance Reporting management
  + Register entrance/exit either manually or by allocator
  + Calculate total hours of each day
  + Reminder for users about their casual entrance
* Monthly attendance Reporting management
  + Generate pdf file contains attendance table of chosen month
* Salary Calculator
  + Calculate user salary by chosen month and year
* Manager management
  + Update employees password
  + Update employees salary

Send notification to proper employee after changing salary

1. **Human Interface Design**
   1. **Overview of User Interface**

* All users :
* Login + registration page

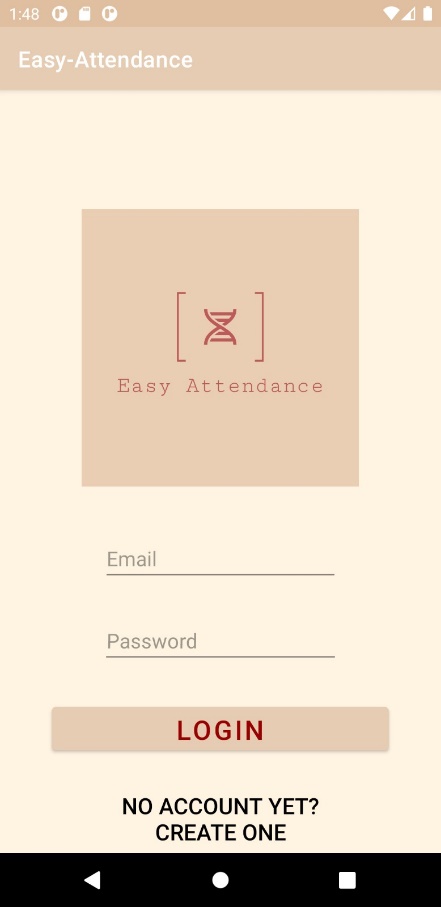
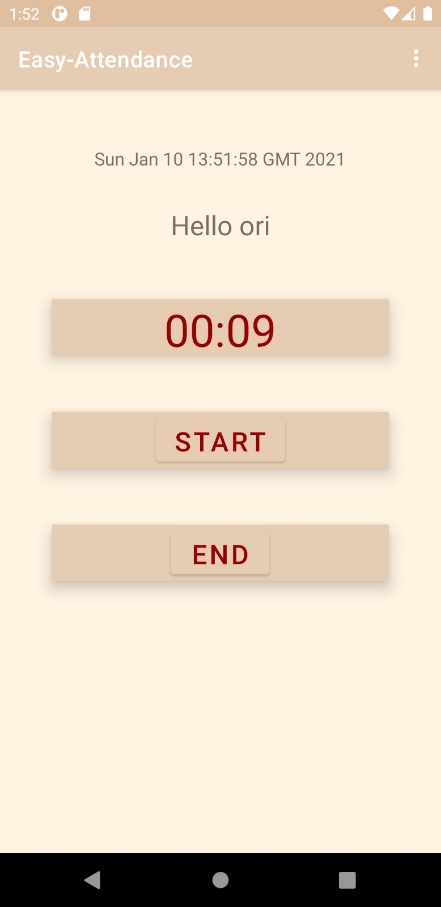
Menu screen:

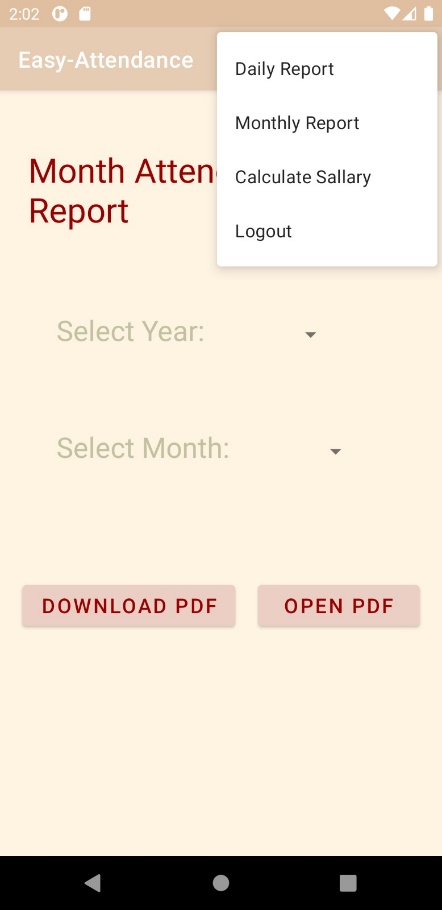
* Daily report
* Monthly report
* Calculate salary
* logout
* Admin :

Addition to menu:

* Workers List

**6.2 Screen Images**





* 1. **Screen Objects and Actions**

Everyone

* Login page
* Registration page

The app has two types of permissions

* Admin – watch a list of employees , update their hourly wage, reset their password, issue the working hours reports of all employees in addition to all the normal actions of a regular user.
* User – Stamp attendance hours, view monthly work reports, payroll calculation .

After the login page, the application will recognize the type of the user and allow the appropriate permissions (user or admin).

1. **Requirements Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Requirements | Permission | Non\functional | Subtype | components |
| 1 | User shall have main screen with 2 button- In, Out.  Pressing on each of the buttons will automatically save the current time in the database. | software | Nonfunctional | Performance | all |
| 2 | One of the menu options will be retroactive attendance. Choosing the option will open new screen that shall give the option to choose a date and fill in/out times for this specific date. The times should be in the format of hour-minutes. | user | functional | data | all |
| 3 | One of the menu options will be generate report. Choosing this option will open new screen. In this screen the user could choose specific month by scroll down menu and then press on another button to generate the PDF. The PDF shall be open on the device. | user | functional | Performance | all |
| 4 | User shall have dedicated screen for salary calculator. The screen should contain textual box for his salary per hour and calculate button. An algorithm should run in the background to calculate the employ salary, taking into account extra hours- 100%, 125%, 150% and 200%. The salary shall appear in popup message. | software | functional | Performance | all |
| 5 | User shall have dedicated screen for viewing his vacation/sickness days status. The user could see how many days he gained and can use, divided by type- vacation, sickness. | software | functional | Performance | all |
| 6 | User could get alerts if he forgot to fill his daily reports as a remainder. This alert should be enabled by configuration in the setting screen. The same alert should be sent to the manager for each of the employs, according to the manager configuration. If employ disable the configuration the manager will still get the alert if he requests to enable the alerts. | software | functional | safety | all |
| 7 | In the left side of the screen there should be a button the by pressing will open menu of all optional modules- salary calculator, vacation days summary, month reports, etc. | software | functional | Performance | all |
| 8 | User could enable automatic fill feature by configuration in the setting window. This feature if he enabled will automatically fill in/out according the coordinates. The coordinates of the workspace will be saved in the DB and if the user enter the area of the coordinates, and this is the first time on this day, the time will be saved as in report. If this is the second time the time will be saved as out record. | user | functional | data | all |
| 9 | While login as manager account the manager shall have the option to view his list of employs, reset their password and update their salary. The main screen will be table of employs, and in each line will be: ID, reset password(button) and change salary. Pressing in each of the button will open the relevant screen/popup | user | functional | Performance | manager |
| 10 | While login as manager account the manager shall have the option to view his employss' attendance reports by using spinners. | user | functional | Performance | manager |
| 11 | The user shall have setting screen, that in this screen he could set the application setting. The user could enable or disable alerts in the screen, or enable/disable automatic fill by coordinate and so on. | user | functional | safety | all |
| 12 | After login, the main screen will contain in the right upper corner menu button. The menu will have list of all optional screen- main, reports, setting, etc. | software | functional | Performance | all |
| 13 | The first screen while opening the application shall be login screen. The login screen shall contain user and password text boxes and login button. In addition, the login screen shall contain another password for creating account in case the user doesn’t have one. | user | Nonfunctional | safety | all |
| 14 | the user could navigate to the register screen from the login screen. The register screen will contain several text boxes for user details- user name, password, email, ID, organization key, etc. The organization key is a unique key per workplace. For each organization key one manager is allowed. To set the account as manager there is dedicated button in the registration key, that will be enabled only if there is no existing manager for this organization key. Under the text boxes a button of sigh up shall appear to finish the registration process. | software | Nonfunctional | data | all |
| 15 | The system shall support all type of devices, and chrome browser. | software | Nonfunctional | Performance | all |
| 16 | The system shall support 2 type of users- regular user(employ) and administrator user(manager). The regular user and the manager user shall differ by their permissions and their optional screens. | software | functional | Performance | all |
| 17 | Users information and reports shall be saved in Firebase DB | software | nonfunctional | data | all |
| 18 | The Firebase DB is not relational DB and save the data in JSON format. | software | nonfunctional | data | all |
| 19 | User shall see and modify only his data according to his user and password and permission. If the user is a manager, he could also see and modify his employs data. | user | functional | Performance | all |
| 20 | The data should be saved in the DB by unique user ID, so the data retrieval should be unique also. | software | Nonfunctional | Data | All |
| 21 | The application will automatically synchronize with the database | software | Nonfunctional | data | all |