

Capstone Stage 1

Description	2
Intended User	2
Features	2
User Interface Mocks	2
Screen 1	2
Screen 2	3
Screen 3	3
Screen 4	4
Key Considerations	4
How will your app handle data persistence?	4
Describe any edge or corner cases in the UX.	4
Describe any libraries you'll be using and share your reasoning for including them.	5
Describe how you will implement Google Play Services or other external services.	5
Next Steps: Required Tasks	5
Task 1: Project Setup	5
Task 2: Implement UI for Each Activity and Fragment	5
Task 3: Add Firebase Cloud Firestore	6
Task 4: Sign In Activity	6
Task 5: Main Activity	6
Task 6: Add Activity	6
Task 7: Widget	6

GitHub Username: LuisVargasVic

Human manager

Description

Keep your employees up to date, have important information and bonus according to each one of them. Forms that integrate crucial fields for each employee.

Intended User

Human resource managers

Features

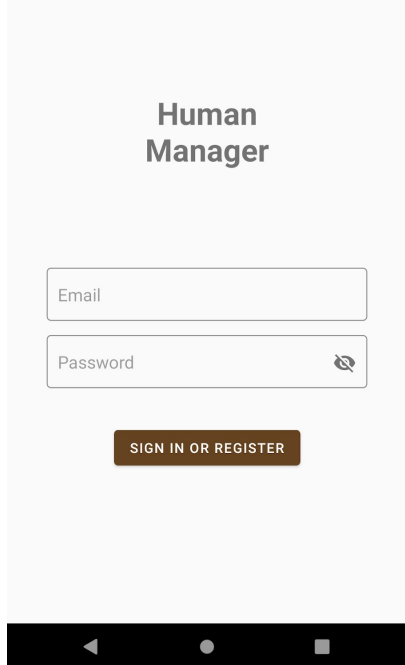
Main features of app:

- Saves information
- Calculate bonus

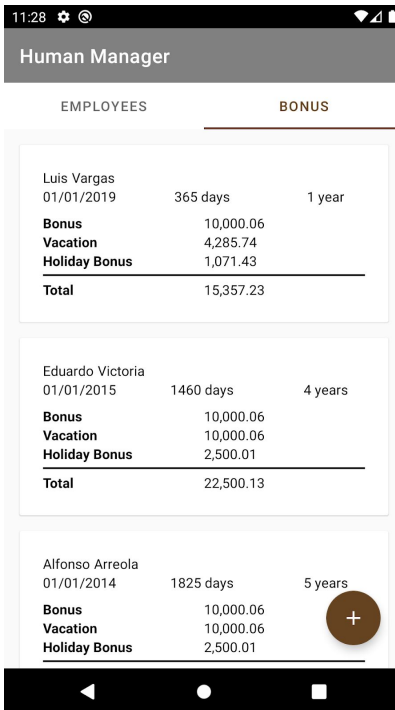
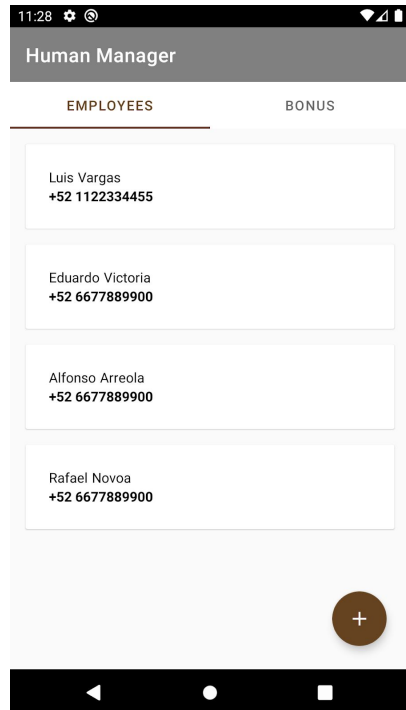
User Interface Mocks



Screen 1

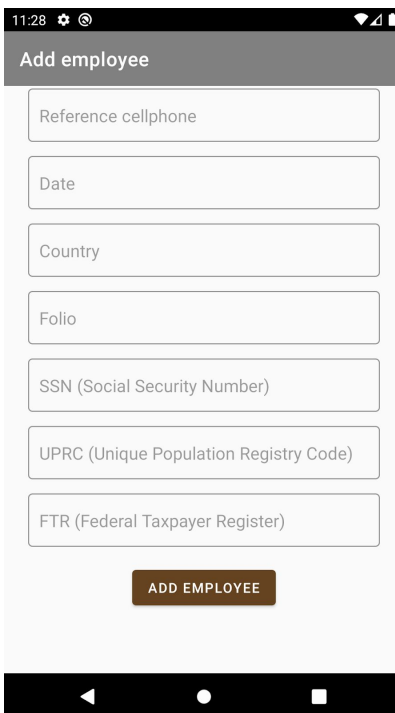
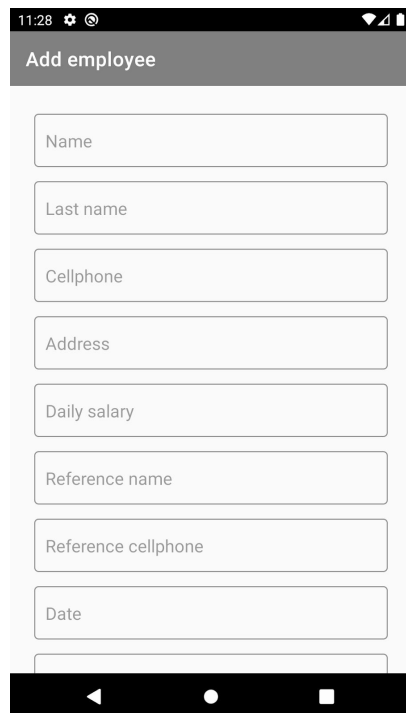


User, in this and only case be a resource human manager, can sign in or register in Screen 1 and when succeed go the Screen 2, otherwise show error message.



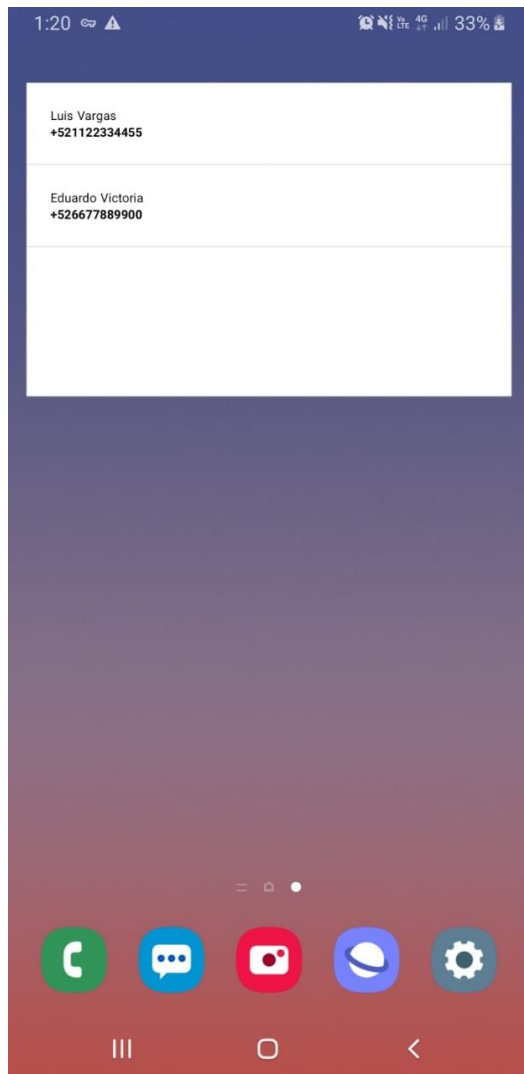
Screen 2

In the Screen 2 main activity will listen to database changes that occur in an update with the employees modified, added or removed.



Screen 3

In the Screen 3 user can add one employee it can only be accessed in main activity 2 with the floating action button



Screen 4

In the Screen 4 user can select employees and see details clicking on the list view of the widget.

Key Considerations

How will your app handle data persistence?

Application will use Firebase Cloud Firestore to store every employer as a document into the resource manager collection of employers

Describe any edge or corner cases in the UX.

Every time user initiates session or sign up for first time back button will exit application because login screen finishes.

Main activity fab shows add activity and after clicking on add button finishes and refresh recyclerview.

For accessibility I will adapt the content descriptions adequate to each element.

Describe any libraries you'll be using and share your reasoning for including them.

App is written solely in the Java Programming Language
Handle database with Room.

```
implementation 'androidx.room:room-runtime:2.2.3'  
annotationProcessor 'androidx.room:room-compiler:2.2.3'
```

Material design

```
implementation 'com.google.android.material:material:1.0.0'
```

Firebase Auth and Firestore

```
implementation 'com.google.firebase:firebase-auth:19.2.0'  
implementation 'com.google.firebase:firebase-firestore:21.3.1'
```

Lifecycle for Live Data

```
implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'
```

Describe how you will implement Google Play Services or other external services.

I will implement google-services for Firebase Cloud Firestore.

In each activity after login I will add Google Admob.

Intent service will be handle by Widget.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Configure SDK version
- Configure Proguard rules
- Configure repository

Task 2: Implement UI for Each Activity and Fragment

- Build UI for SignInActivity
- Build UI for MainActivity
- Build UI for EmployeeFragment

- Build UI for EmployeeAdapter
- Build UI for BonusFragment
- Build UI for BonusAdapter
- Build UI for AddActivity

Task 3: Add Firebase Cloud Firestore

Add firebase cloud firestore to google-services.json expecting it in .gitignore, also add Room to adapt the database

- Implement database room
- Implement firebase cloud firestore

Task 4: Sign In Activity

Implement authentication into Sign In Activity

- Implement functionality
- Implement Firebase Authentication

Task 5: Main Activity

Implement listeners for each users collection called “employees” update every time a document changes.

Sign out remove data from database and end firebase session.

Ad banner implemented.

- Implement functionality

Task 6: Add Activity

Add Employer into database and also use this activity to update each Employer.

- Implement functionality

Task 7: Widget

Implement widget to access to all the employees that are on the database.

- Implement functionality