How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: shimonaj

HelpDesk

Description

This is an app for the customers of a support group.

For example :A residential society offers an app for it's customers so that they can raise a ticket for household support issues, like "Need electrician" , "Water pipe broken" etc.

As the customer logs-in , he can view the status of all the tickets raised by him and add additional comments or close the ticket.

Intended User

Customers of a Support Company

Features

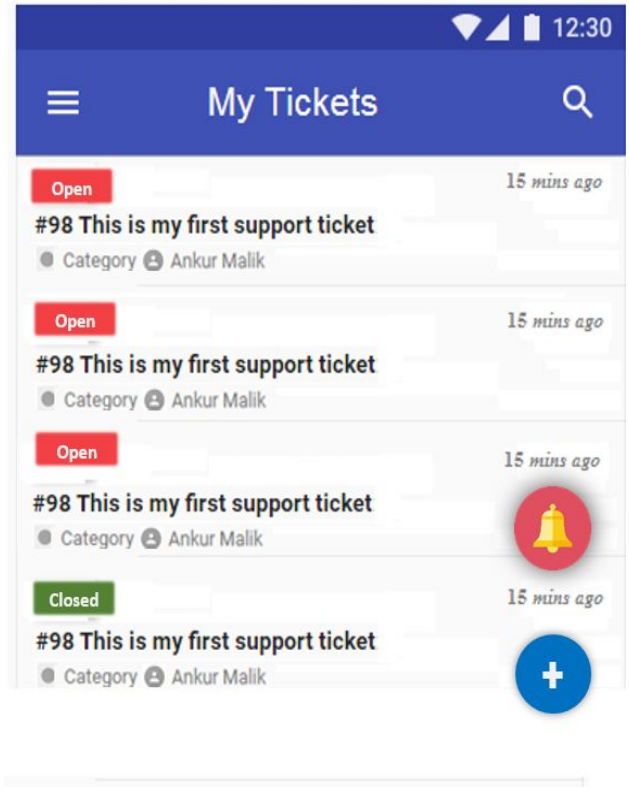
- Create Ticket
- View Status of tickets
- Reply on the ticket
- Close Ticket

User Interface Mocks

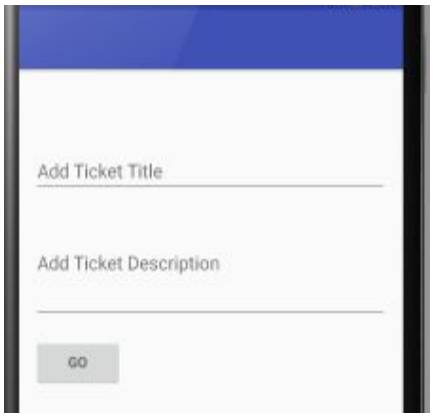
Screen 1: This is the first screen, Customer has to mention the Company Name (provided to him) to log-in the system

Screen 2: Login Screen

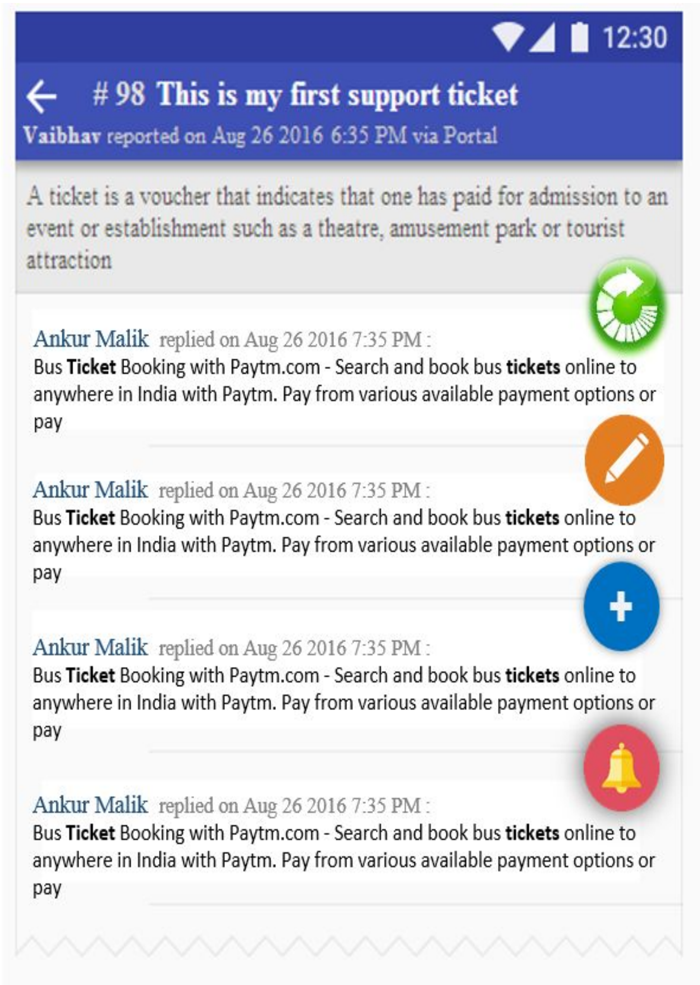
Screen 3: This will be the interface were all the tickets raised by customer will be displayed.(Please note I'll have only the Create Ticket "+" Fab button, Rest will not be there in actual App.)



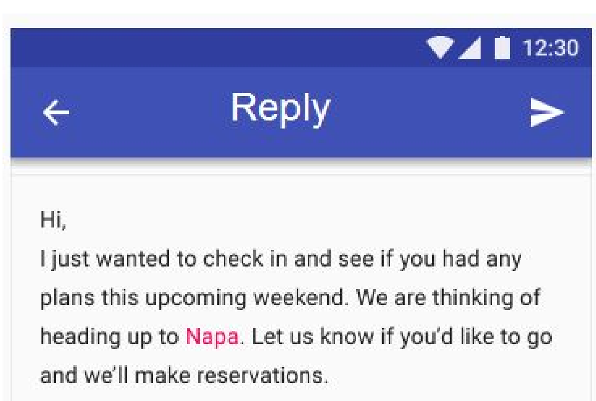
Screen 4: This is the Create ticket interface



Screen 5 : This will open up, when we click on a ticket Tile. (Please note I'll have only the Add reply "+" Fab button, Rest will not be there in actual App.)



Screen 6 : This is the interface to add additional comments on a ticket. This will have menu options : Reply, Reply and Close,Cancel.



Key Considerations

How will your app handle data persistence?

I'll create two tables for the offline storage, Ticket table and the Comments relational table.

Describe any corner cases in the UX.

None

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

Recycler View, Card Views are used to display the tickets
OkHttp to connect with the backend web-services

Describe how you will implement Google Play Services.

Not required.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Expose the webservice methods to interact with the backend database like:

- validateCompanyName
- validateUser
- onLogout
- fetchTickets

- fetchTicketComments
- postTicket
- postComment
- getTicketDetailById

Task 2: Implement Content Providers

- Implement Content Providers for Ticket Table, and Comments table

Task 3: Implement Remoting Services

- Implement Services which will contain the common methods to interact with the backend rest based services
- Implement the Updater Intent Service to perform all the tasks like fetching the tickets from backend and updating the sqlite db as well as broadcasting the event status.

Task 4: Implement UI for Each Activity and Fragment

Build UI for:

- LoginActivity
- MainActivity
- CreateTicketActivity
- TicketDetailActivity
- TicketDetailFragment
- TicketCommentFragment
- ReplyActivity

Task 5: Your Next Task

- Unit Testing
- Integration Testing

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"