GitHub Username: hmunguba

Condominium

Description

Use Condominium to keep track of everything that is happening in your community!

Condominium app can help you check if your community's common areas are available and schedule an event. Receive notifications when your mail arrives and many other announcements about your condominium.

How it works?

It is easy! First of all, make your registration to the community you are part of (if the community does not exists in the app yet, talk with the responsible for the community to create an account) and then you can use it.

Invite your neighbors to register themselves too. The more people in the community use the app, the better!

Intended User

This application is developed in Java and is intended for people living in community. People who live in a condominium of houses or apartments. People who share common areas like pool, party room, barbecue area, etc. with the neighbors.

Is also intended for people who would to know when your mail arrived without having to call to the concierge, or even send notifications to the concierge when you are not at home.

n general is for everybody who lives in community.

Features

- Access information
- Saves information
- Send notification
- Receive notification
- Register user
- Make call
- Send message

User Interface Mocks

Screen 1

Login Screen – This is the initial screen. The user must login to have access do the application.

Screen 2

Profile screen – When user logs in (only for the first time) he must choose a profile: person or condominium, then he is redirected to edit the profile of the chosen option.

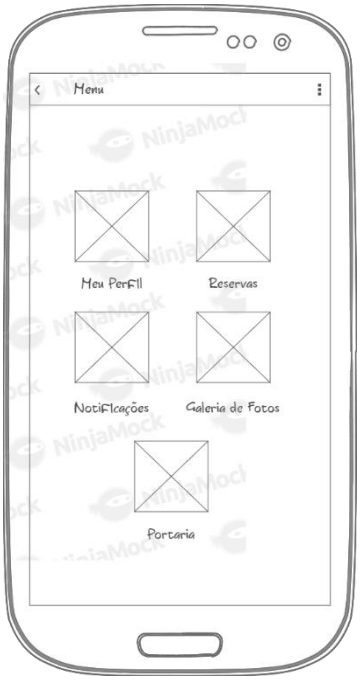
Screen 3

My Profile (User) – User may set a profile picture (optional) but he must set his name, apartment, city and condominium name. The condominiums available are displayed in a spinner, so it must be previously created.

Screen 4

My Profile (condominium) – User may set a profile picture (optional) but all the other items (name, localization, state, city) must be set. If the user clicks in “Visualize more items” he must add other information about the condominium such as if has a pool, barbecue are, gourmet area, etc.

Screen 5



Menu Screen – Display all available options:

- My Profile: where user may edit the profile
- Events: where user may see all events in agenda and create a new event
- Notifications: check all notifications received
- Gallery: visualize all condominium pictures
- Concierge: Call or send message to lobby

The menu screen is the same for both user profile and condominium profile.

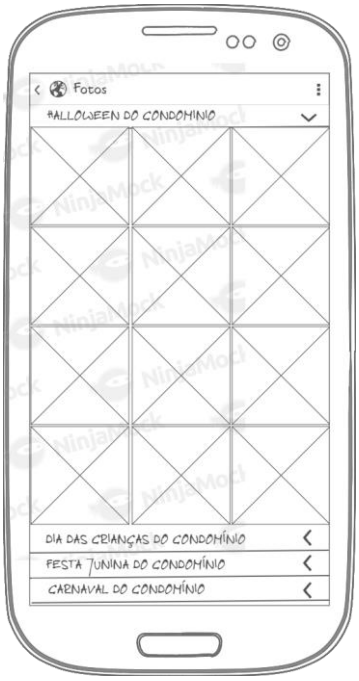
Screen 6

New Event Screen – User may create an events or reservations to make private use of some common areas of the condominium for a period. All displayed fields must be filled to create a new event.

Screen 7

Notifications Screen – Display all sent and received notifications.

Screen 8



Gallery Screen – Display pictures separated by albums. Those pictures may be added by users or by the condominium.

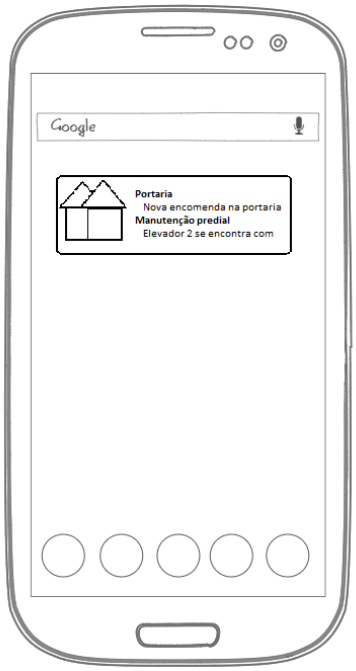
Screen 9

Concierge Screen – Display options for user to Call or to Send Notification to lobby.

Screen 10

Send Notification Screen – Display options to send notifications. It must include title, recipient and content.

Widget UI



Widget – Display the application's icon and the latest notifications received. When widget is clicked, it displays the Notifications screen.

Key Considerations

How will your app handle data persistence?

Cloud Firestore will be used to store information of each of the condominiums and users registered.
To a condominium be available in the app, first of all it must be created a condominium profile, once it is created, the server will display this option for the users who will be registered.

Describe any edge or corner cases in the UX.

If the user exits My Profile Screen without creating it (pressing the Enter button) then the user will not be effectively created in server, however, if the activity is not destroyed, all the information will be maintained in the activity. The same rule is applied for all the others activities in the application.

When user have no connection to the internet he cannot create a profile (because the condominiums available list depends on internet connection). The other actions like send notification or register an event, will be sent to server as soon as user have connection to the internet

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

- EasyCalendar to customize the calendar to show and create events.

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

- Firebase Authentication to register users.
- Firebase Realtime Database to check for condominiums, events, photos and notifications.
- Cloud Messaging to send notifications to users.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create github repository
- Configure libraries
- Create project in Firebase
- Create Firebase realtime database
- Setup Firebase authentication

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for ChooseProfileActivity
- Build UI for RegisterUserActivity
- Build UI for RegisterCondominiumActivity
- Build UI for MenuActivity
- Build UI for EventsActivity
- Build UI for EventsFragments
- Build UI for NotificationsActivity
- Build UI for NotificationsFragments
- Build UI for GalleryActivity
- Build UI for GalleryFragments
- Build UI for ConciergeActivity
- Build UI for ConciergFragmetns

Task 3: Create Image Resources

- Create application launcher icon
- Create icons
- Select application main colors
- Apply theme to activities and fragments

Task 4: Implement Firebase Services

- Implement user authentication
- Implement cloud message
- Implement realtime database

Task 5: Implement Application Actions

- Create / Edit / Exclude / Visualize events
- Create / Exclude / Visualize notifications
- Create / Edit / Exclude / Visualize pictures from gallery

- Develop call feature
- Develop send message feature
- Implement SyncAdapter to update data

Task 6: Develop tests

- Develop unit tests to validate input
- Develop instrumented tests

Task 7: Develop Widget

- Develop widget layout
- Develop widget provider
- Develop widget service
- Display notifications in widget
- Test if notifications are being update correctly in widget

Task 8: Handle Error Cases

- Handle error cases for profile
- Handle error cases for events
- Handle error cases for notifications
- Handle error cases for gallery
- Handle error cases for concierge services
- Handle error cases for widget

Task 9: Necessary Tasks

- Implement accessibility to all screens
- Move all texts to strings.xml file
- Sign apk