
GitHub Username: Veskol

Wisher

Description

Wisher is a mobile app that allows you to create present Wish-lists for all kinds of occasions and celebrations. Free and easy to use, you can tell your friends and family what presents you'll really enjoy, making their choice easier and your life happier.

Why use Wisher:

- Receive the gifts you really want and will make you happy.
- Save time and nerves to your close people, by letting them know what exactly do you want.
- Never receive the same thing twice (no one needs 3 identical coffee-sets).
- Easily share your wishes to your closest people.

Just create a list and have a great time (and presents :D)!

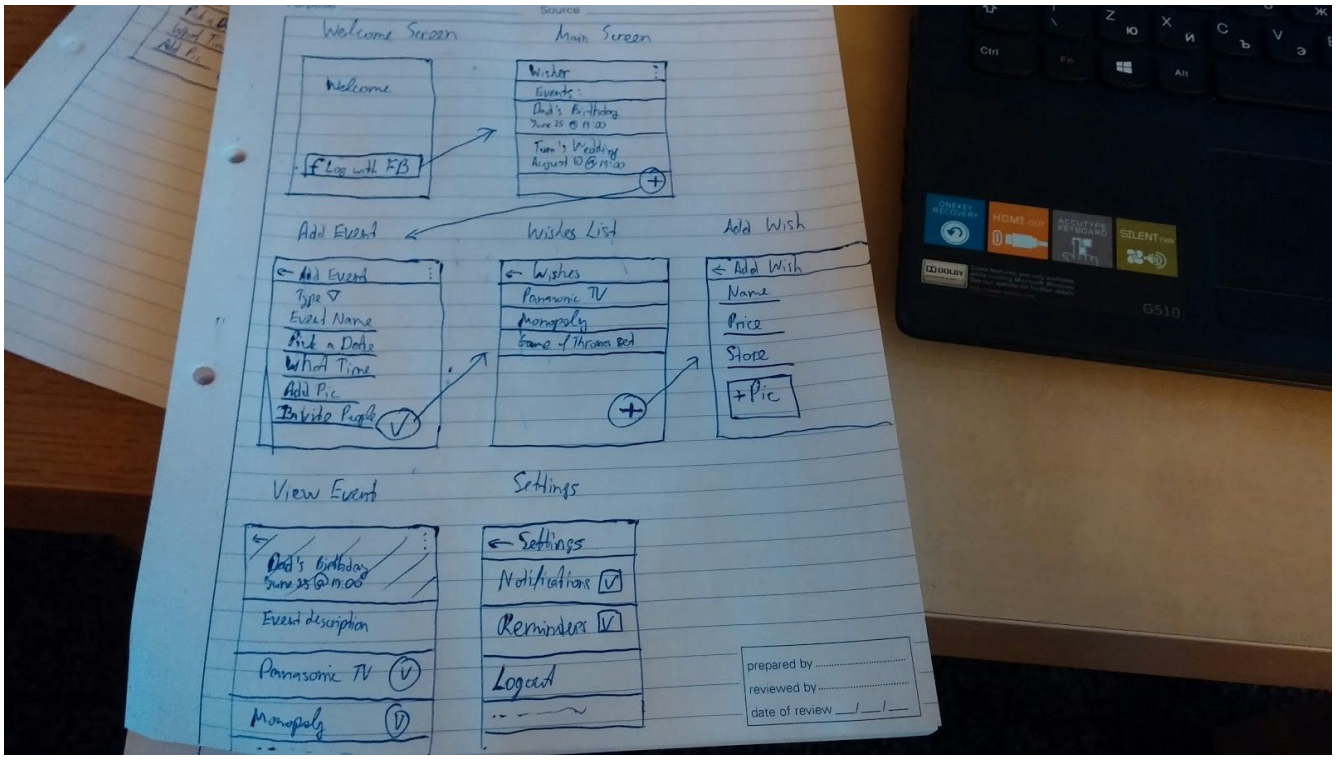
Intended User

Target group for this app are modern people that like having fun, celebrating with friends and family and receiving gifts … or stated in other way - **everyone** :) Perfect for creating a present wish list for a birthday, wedding, name day, anniversary and all other occasions imaginable.

Features

- Create an event or join an existing one you're invited to
- Add presents to your wish list for events you're organising
- Guests mark when a present is bought, so that no other gets the same thing
- All information is saved in the Cloud (events, items, etc), thus all users are always up-to-date
- Notifications when a present wish is changed
- Reminder of an upcoming event, so you don't forget about it

User Interface Mocks



Welcome Screen
Provides just a Login / SignUp button. Only way to enter the app (for now) is Facebook login.

Main Screen
Holds a list (RecyclerView) of “your events” - the ones you’ve created and those you’ve been invited to. The FAB allows you to create a new event, redirecting you to **Add Event** screen. Clicking on an existing event brings you to **View Event** screen.

Add Event
Provides appropriate input fields for all needed info to create an event. The “save” FAB action takes you to **Wishes** screen.

Wishes
Provides a list of all registered wishes for this event. Clicking an item displays more details about it. The FAB redirects you to the **Add Wish** screen.

Add Wish
Provides appropriate input fields to add a wish to your list. Upon saving you’re redirected back to the **Wishes** screen.

Key Considerations

How will your app handle data persistence?

The app will be build around the Parse BaaS, so it’ll use ParseDatastore and ParseLocalDatastore to persist all info.

Describe any corner cases in the UX.

The UX flow and navigation will be as clear and straight-forward as possible, meaning that you’ll always start from the Main Screen (or Welcome Screen if not logged in), and navigate down (and back up) the hierarchy from there. For example: Main Screen ->View Event -> Item, back to the Event, back to Main Screen.

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

1. Parse SDK, which will handle all communication with the backend, image loading, uploading, push notifications and more.
2. Android Design Library to easily make use of the latest Material design.
3. ButterKnife to shorten the UI initialisation code required.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project
- Add ParseSDK, ButterKnife and Android design library dependencies

Task 2: Implement UI for Each Activity and Fragment

- Add a BaseActivity and BaseFragments classes to hold common logic that must be shared for all Activities/Fragments
- Build UI for Welcome Screen
- Build UI for MainFragment, adding some dummy data to the events list to verify the visuals are working properly
- Build UI for AddEventFragment
- Build UI for WishesListFragment
- Build UI for AddWish Fragment
- Build UI for ViewEventFragment
- Build UI for Settings screen, using a PreferenceActivity

Task 3: Prepare Parse backend (to be done from Parse admin panel)

- Create / edit datastore models for User, Event, Wish
- Prepare some push notification tests

Task 4: Utilise Parse SDK in the Android project

- Add needed code to support user login. Hook up the “Log in with FB” button on the main screen, using Parse authentication methods
- Fetch Events for MainScreen from Parse
- AddEventFragment should add the Event to Parse datastore
- AddWish screen should add the Wish to Parse datastore
- ViewEvent should pull data from Parse datastore
- Consume Parse Local Datastore - enable it and load everything from there first
- Allow for Parse push notifications
- Allow Parse analytics in the app (add tracking for key events in the UI flow)

Task 5: Revise UI to fully support Material design and tablets

- Implement a “Master-Detail” flow using the fragments already in place
- Ensure the app looks good both on all devices (phones and tablets)