GitHub Username: MelanieJae

Dignity Memorial(R) Mobile Application

Description

This app is an adaptation of an existing website for Dignity Memorial, LLC, a network of funeral planning and site providers throughout the US. It provides quick access to the most valuable services that the existing Dignity Memorial website provides, e.g. search of obituaries, search for service providers (e.g. burial, cremation), a “Things to Remember” checklist, and creating, saving and sharing an event plan via a mobile device.

Intended User

Anyone who is pressed for time and needs to plan a memorial and burial or cremation for a loved one as well as those attending services or the burial site of a deceased family member or friend in another part of the country.

Features

- Search of obituaries
- Search for service providers (e.g. burial, cremation)
- “Things to Remember” Checklist
- Scheduling a consult for pre-planning/financing OR Create, save and share a memorial and burial/cremation plan through phone
- Find/navigate to a loved one's memorial and/or burial site gives Google Maps directions and, when geofencing is activated, continues right to actual plot
- Bill Pay

User Interface Mocks

These are present in a separate document linked in the README for the Github repository titled “Capstone_Project”. The mocks provided are for any screens which do not simply display the web browser and the existing company website (i.e. the bill pay and checklist screens)

Key Considerations

Data persistence

User selections in Settings (including any and all plan choices) will be stored to a local DB which is accessed via a custom content provider that I will create.

UX and its corner cases

App will use typical navigation (e.g. back button, overflow menu button and navigation fragment with menu options). Overflow menu will be accessible from every screen and contain all of the same options as the menu fragment as well as a Settings item where user preferences will be stored and retrieved as needed throughout the app.

Third party libraries

- Picasso and/or Glide to handle the loading and caching of images.
- ButterKnife to handle UI view injection to reduce multiple calls to “findViewById”
- Timber to ease logging and exception handling since there will be many user input fields and opportunities as well as need for null checks and exceptions in the content provider.

Google Play Services implementations

- Maps-for location of service and/or burial site to guide user to the location
- Analytics-track page views, possibly also tags
- Location/Context-used to change navigation format from Google Maps to locating precise burial site or mausoleum within cemetery

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

- Gradle dependencies and/or library installations
- Gradle installRelease and any other custom tasks
- Application Java directory and class for creating one-time instances of Google Play services used throughout the app

Task 2: Implement UI for Each Activity and Fragment

- Create master/detail flow activities/fragments
- Master Fragment:
    - Complete navigation menu items (layouts and classes)
    - Complete navigation logic to detail screens for each item (e.g. intents)
- Detail Fragments:
    - Add detail fragment classes for each menu item
    - Customize detail fragments to particular screen

- Create UI directory:
    - any custom RecyclerView holders or adapters
    - Empty views
- Create overflow menu that will be accessible throughout app

Task 3: Data Persistence

- Create schema for user settings database table
- Create Data directory: contract class, custom provider and DBHelper classes
- Create settings activity: this is what database structure will be for in this app; provider will be accessed and used for CRUD functions for all user settings (e.g. choice of service provider, choice of burial site, event planning) throughout app and on widget

Task 4: Data access handling during app operation

- Create IntentServices and/or AsyncTasks for the search menu items' detail activities
- Create loaders to link database information with settings and load info obtained from web services into UI elements

Task 5: Implementation of Google Play Services

- Implement Location/Context in "Find service or site" menu item detail fragment
- Implement Analytics for page view tracks and possibly tags in all screens (via application class)
- Implement Maps for "Find a provider", "Find obit" and "Find service or site" detail screens.

Task 6: Accessibility Features

- Implement Spanish language support
- Implement RTL layout support
- Implement Talkback support
- Implement peripherals support (e.g. D-Pad navigation)

Task 7: Widget

- Create widget displaying event and location details, or if none, a “need to plan a memorial?” link that opens the app (i.e. “empty view”)
- Implement Analytics for page view tracks and possibly tags in all screens (via application class)
- Implement Maps for “Find a provider”, “Find obit” and “Find service or site” detail screens.

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**”