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: MattPflance

HearthList (TBD)

Description

HearthList allow you to easily view all cards in Blizzard's game "Hearthstone: Heroes of Warcraft" and create custom decks that you can save on your device and share with your friends. HearthList downloads the latest data on your phone so you can view content offline.

Intended User

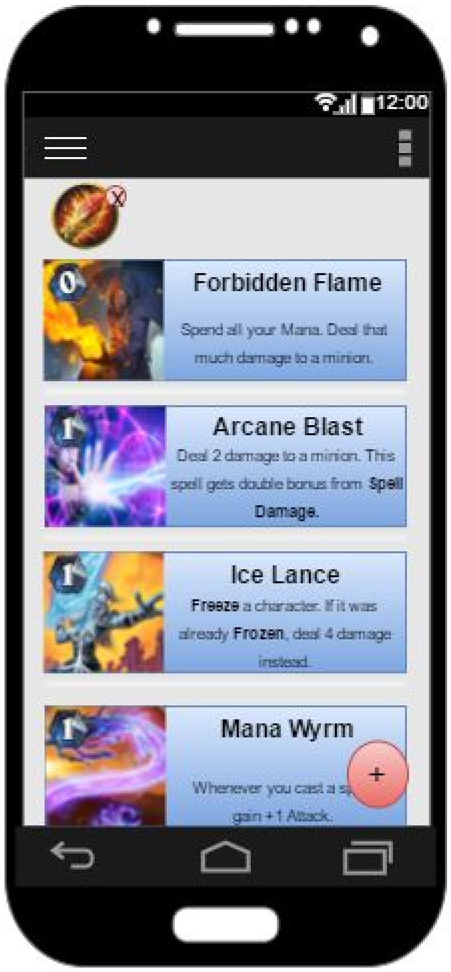
Hearthstone players.

Features

- See list of all available Hearthstone cards
    - Filters: Mana cost range, Class, Attack range, Durability range, Rarity
    - Art, details, card backs
- Build and save custom decks with Deck Builder
- Share decks
- Offline Mode available after initial setup.
- Available for tablets and phones.

User Interface Mocks

Screen 1



0 Forbidden Flame
Spend all your Mana. Deal that much damage to a minion.

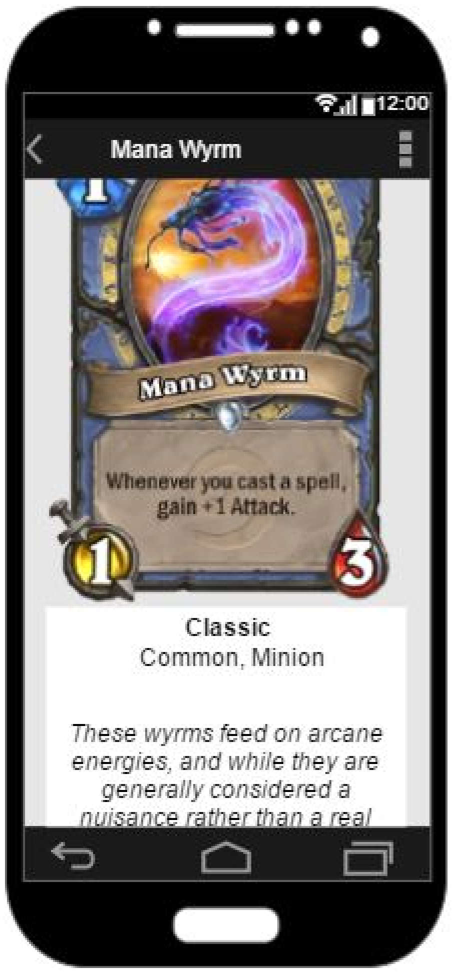
1 Arcane Blast
Deal 2 damage to a minion. This spell gets double bonus from **Spell Damage.**

1 Ice Lance
Freeze a character. If it was already **Frozen**, deal 4 damage instead.

1 Mana Wyrm
Whenever you cast a s
gain +1 Attack.

Filter through all available cards.
Scroll horizontally through filters; delete and add as you wish.
Easily determine Class cards by Class color.
View compact version of cards as a vertically scrollable list.

Screen 2



Classic
Common, Minion

These wyrms feed on arcane energies, and while they are generally considered a nuisance rather than a real

Shows image of selected card (toggle between gold and normal), card set, rarity, type of card, flavor text, and other stats.

Key Considerations

How will your app handle data persistence?

The app will pull data using a Hearthstone API from Mashape on initial launch. It will store all data into a Content Provider and load from there. When new cards come out, I will initiate the app to make another API request (via Tag Manager) to get the newest data. Content will always be available offline after initial launch and app will be prompted to connect to internet to update cards when new cards are available.

Describe any corner cases in the UX.

- Navigation Drawer will load main content onto the FragmentManager backstack and rotating device will have to keep track of current Fragment’s title.
- Selecting a card will remove navigation “Hamburger” icon and replace with the Up button.
    - No other action will remove the navigation hamburger icon

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

- Glide for image caching and loading from the Hearthstone API
- Google Analytics for tracking and Tag Manager for dynamic updates
- Google AdMob for ads

Next Steps: Required Tasks

Task 1: Project Setup

- Draw out relationships between different components of app
- Determine Classes and Objects to work with
- Determine how to setup CoordinatorLayout
- Sketch all screen UI

Task 2: Implement ContentProvider

- Write Contract
- Determine variables for each card
- Provide paths for queries

Task 3: Implement Classes and Objects

- Card, Deck, etc
- Implement as Serializable for easy storage in database

Task 4: Implement UI for Each Activity and Fragment

- Build UI for MainActivity

- Build CardList Fragment
- Build CardDetail Fragment
- Build Filtering UI (Horizontal scroll)
- Build DeckList UI

Task 5: Add Google Play Services

- Register project in Console
- Set up Admob
  - Add Test Ads
- Set up Analytics
  - Add tracking and Handler for Tag Manager dynamic updates (for new cards)

Task 6: Implement CardList/CardDetail interaction

- Add smooth transitions and animations
- Load correct card when selected

Task 7: Implement Filtering system

- Set up horizontal scrolling for filters
- Add filters (FAB) on all cards (class, mana range, minion/spell, mech/pirate/murloc, etc)
- Removable filters by tapping active filters

Task 8: Design Deck Builder / Decks Viewer

- Add multi-pane fragments of card list and current deck state (and current saved decks)
- Save decks to ContentProvider in separate table
- Display list of saved decks when accessing decks

Task 9: Make app Accessible, Localized, and Support RTL layouts

- Add Content Descriptions to images
- Make app support left-to-right and top-to-bottom navigation
- All strings in strings.xml
- Add Right/End, Left/Start to xml layouts

Task 10: Support Tablets and Phones

- Optimize all UI for tablets
- Use Multi-Pane Fragments when appropriate