November 1, 2015

DPS924 Assignment 1

Minna Lin
mlin56@myseneca.ca

Contents

Executive Summary

The proposed Android app is an equipment planner and optimizer for the online game *Final Fantasy XIV* (FFXIV). Presently, the only tool that exists for doing this is a web app that is poorly optimized for mobile users, so the need for a mobile-friendly app does exist. Features proposed include a searchable item database, a gear set builder and optimizer, a personal inventory tracker, and the ability to save custom sets. The project will be completed by the end of Week 12 of the Fall 2015 semester.

Reasons for Mobile

A website currently exists (http://ffxiv.ariyala.com) that performs the planned functions of this app, and it is one of the most popular FFXIV resources. However, it is not optimized for mobile devices at all – the wide layout isn't responsive, hover features don't work, the bar that's supposed to be fixed to the bottom of the screen doesn't work, etc. One of its most useful features, the "Best-in-Slot Solver", displays a box in the middle of the screen, but on mobile browsers it aligns strangely, and the text fields and checkboxes are difficult to work with because of their small size. A mobile app is the ideal solution because it can take advantage of touchscreen interaction types (long press, slide, etc.) to handle the UI issues that the website has.

Business Case

This app would be desired by those who would like to plan their equipment upgrades on mobile devices with an easy-to-use mobile-friendly interface. This could be people who would like to work on it while away from home, or PS3/PS4 players who don't have a PC in the same room as their console.

Features

Item Database – The app will include a database containing all currently relevant equipment and items. Initial data will be added to the database using the XIVDB API (http://v2.xivdb.com/api). The user may view individual entries or filtered search results.

Set Builder – Similar to the main interface on Ariyala. After selecting their class, the user can customize their equipment set slot by slot and the total stat values will be visible at all times. Items visible on these screens can be filtered by level, rarity, source, etc.

Set Optimizer – Same as the BiS Solver on Ariyala. Given a set of stat weights and requirements, will automatically determine the best possible set of equipment for that class.

Inventory – The app can track which items the user currently has access to, and calculate an optimal set based on their personal inventory.

Saved Sets – The user can save multiple equipment sets.

Users

The users of this app will be FFXIV players who want to optimize their stats. The fact that they're spending the effort to do so means that they're most likely already familiar with terminology, so game-specific terms can be used without having to worry about confusing them.

Use Cases

Search:

1. The app displays a search bar and filter options.
2. The user fills in their search terms and hits enter.
3. The app displays a list of items that match the search terms.
4. a. The user clicks an item to view details.
   b. The user long-clicks an item to add to inventory.

Build Set:

1. The app displays a list of classes and races.
2. The user selects their class and race.
3. The app has a row of tabs representing each equipment slot, and displays a list of items for the selected tab. Total stats based on selected equipment are displayed at the bottom.
4. a. The user clicks an item to select it, updating the totals shown.

   b. The user long-clicks an item to view details.

   c. The user clicks a tab to switch to another equipment type.
5. The user selects *Save set* from the options menu, and enters a name when prompted.

Optimize Set:

1. The app displays a list of classes and races.
2. The user selects their class and race.
3. The app displays a form for stat weights and accuracy rating requirement.
4. The user enters in the data.
5. The app calculates and displays the best possible set.
6. The user selects *Save set* from the options menu, and enters a name when prompted.

Load Set:

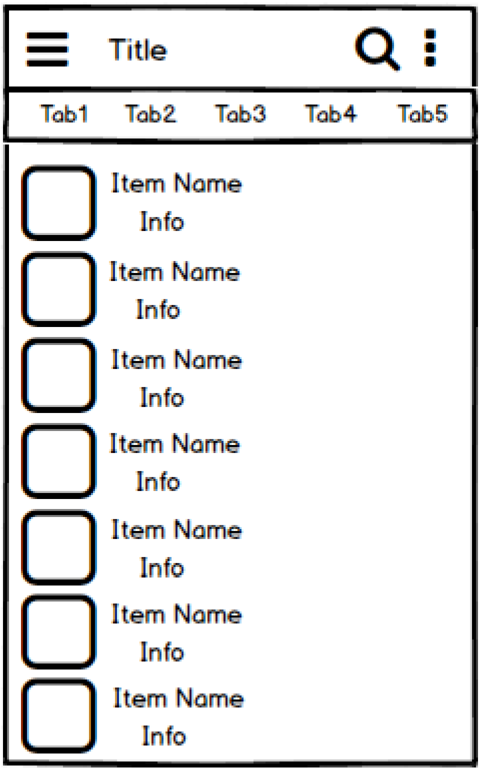
1. The app displays a list of saved sets.
2. The user clicks on the set they wish to load.
3. The app transitions to the set builder with the set items pre-selected.

View Inventory:

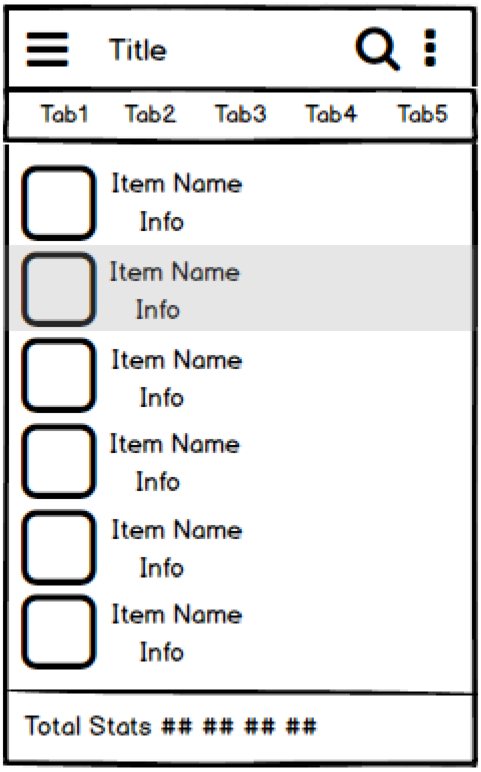
1. The app displays a list of classes.
2. The user selects a class.
3. The app displays a list of all items in the user's inventory usable by that class, with a row of tabs at the top to filter by type.
   a. The user clicks an item to view details.
   b. The user long-clicks an item to bring up a context menu with the *Remove* option.
   c. The user clicks a tab to filter the item list.

View Mockups

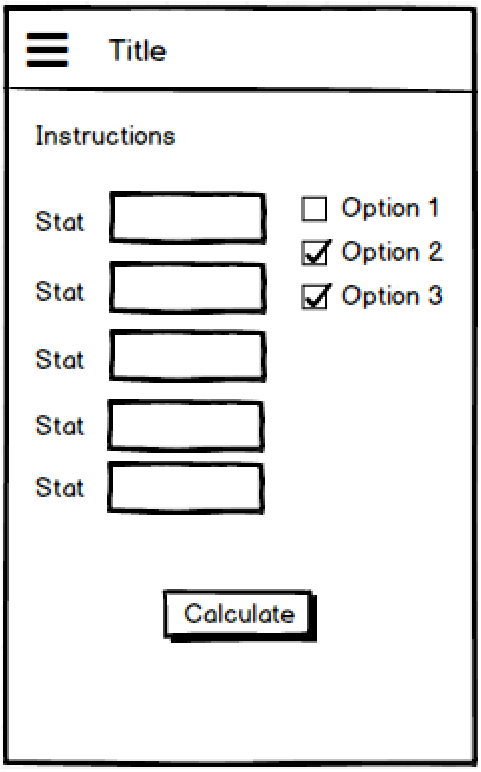
Inventory



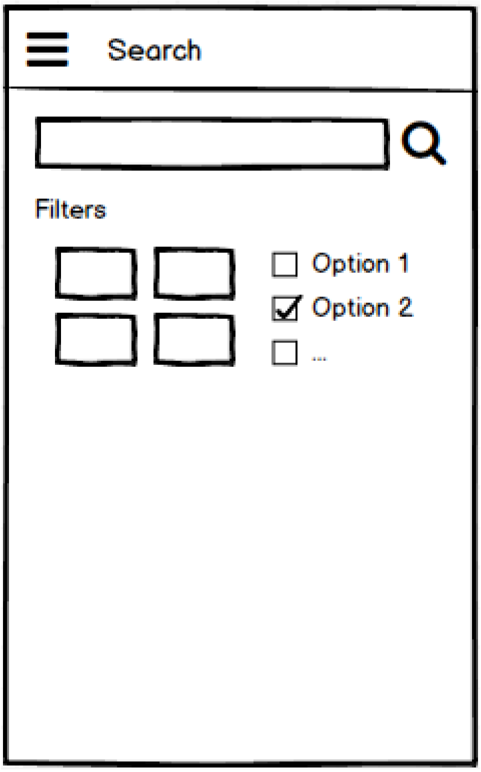
Set Builder



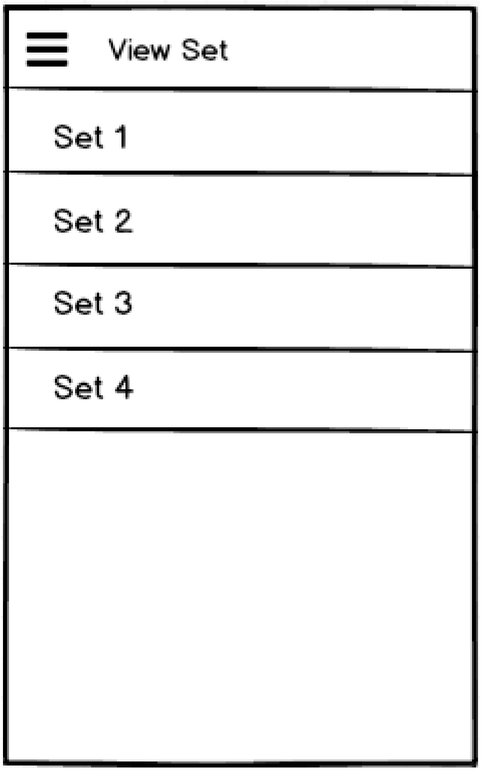
Set Optimizer



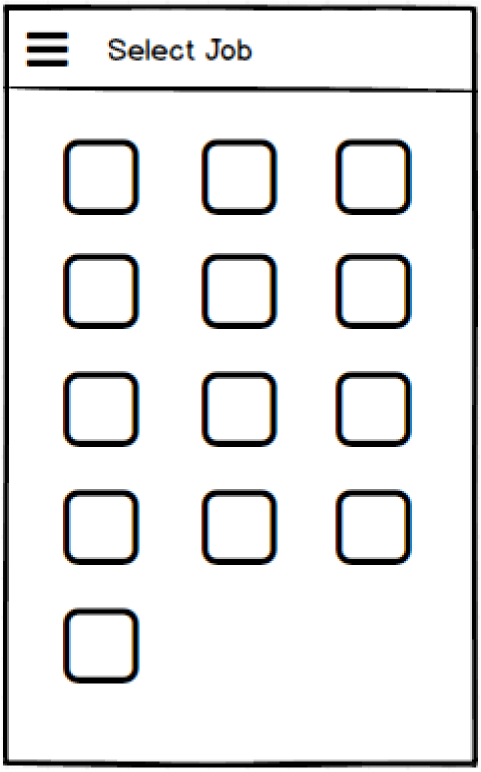
Search



Set List



Class Selection



Timeline for Completion

Week 8 – Create basic app that loads item database and has a working search function

Week 9 – Create the Set Builder screens

Week 10 – Implement Inventory and Set Optimizer

Week 11 – Implement Saved Sets

Week 12 – Clean up and write document