

Penn Course Review Android Application

Customer Handoff Document

CIS350 - Software Engineering & Design

Jinyan Cao, Connie Ho, Charles Kong, Cynthia Mai

Client: Amalia Hawkins, Kyle Hardgrave (PennApps Labs)

Project Lead/TA: Zach Zarrow

Project repository: <https://github.com/jinatonic/Penn-Course-Review-Mobile-Beta>

User Manual

1. Authentication Page

Upon the user's first time using the Penn Course Review application, the first screen seen is the Authentication Page. The user must be a current Penn student with a valid PennKey to access the application. Tapping on the left button, Get Serial Number, opens up a browser on the user's phone (Figure 1b.) The user must first enter his/her PennKey and password to retrieve a serial number. The user can then highlight and copy the serial number, press back on his/her phone to return to the Authentication Page, and paste it into the text box. The user then can press the right button, Authenticate, to authenticate his/her serial number (and thus Pennkey). Upon successful authentication, the Start Page (section 2) is brought up. Upon future uses of the application, the first screen seen will then be the Start Page rather than the Authentication Page. The user's serial number is saved locally on the user's phone, so the user only needs to authenticate him/herself once. Upon failed authentication, a toast appears stating that authentication failed and prompting the user to try again.

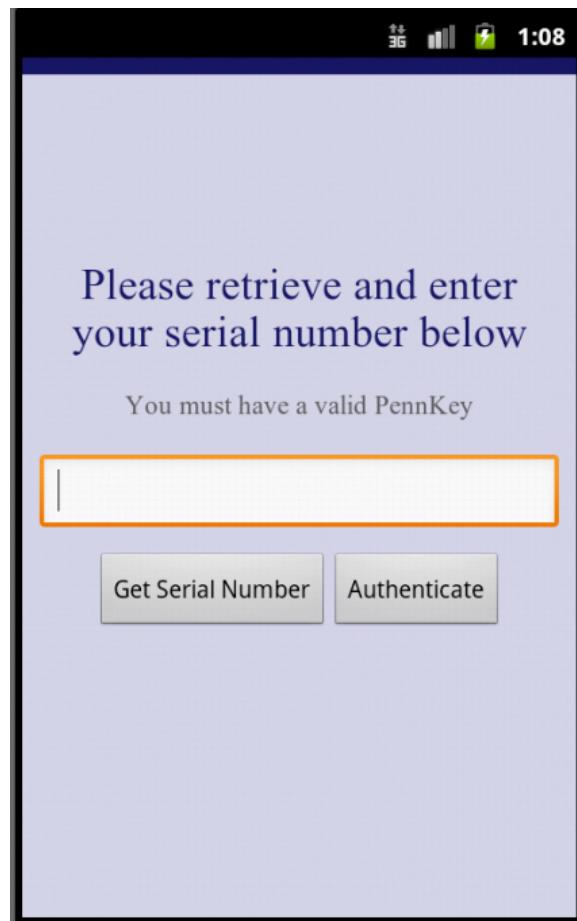


Figure 1a. Authentication Page

1.1 Serial number retrieval from browser

Here the user can select and copy his/her serial number, as stated above in section 1.

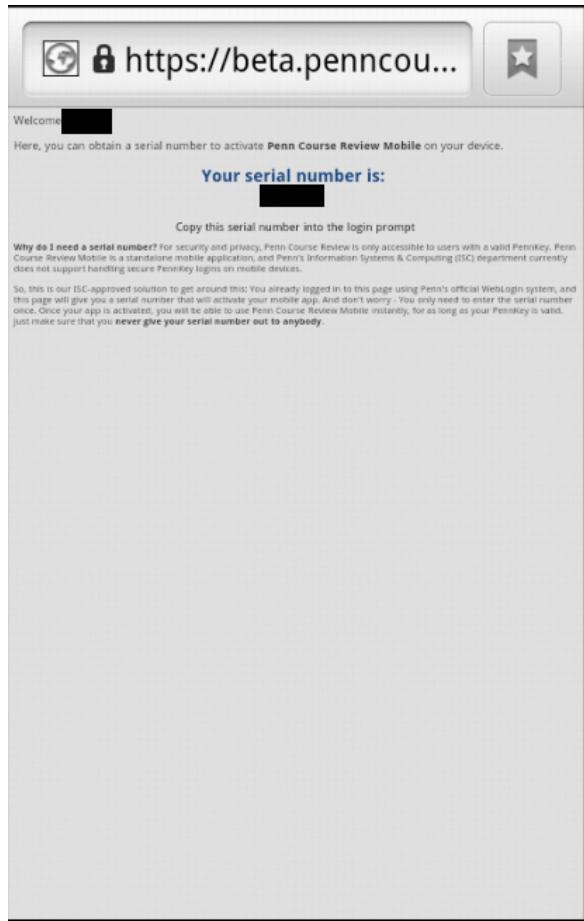


Figure 1b. Browser for serial retrieval

2. Start Page

2.1 Autocomplete loading

On the first time a user uses the application, the autocomplete database immediately starts downloading. Depending on the user's connection speed and stability, the average download time is around 30-40 seconds on strong wireless or 4G, and around 45 seconds on 3G. On future uses of the application this information is already stored locally on the user's phone and so the first screen brought up is the Start Page (Figure 2b). As autocomplete downloads, the user can see the percentage and fraction of departments left to download. The currently-downloading department can also be seen. The only other time this screen is returned to is when a user decides to clear the autocomplete database from the Settings page (section 9.2).

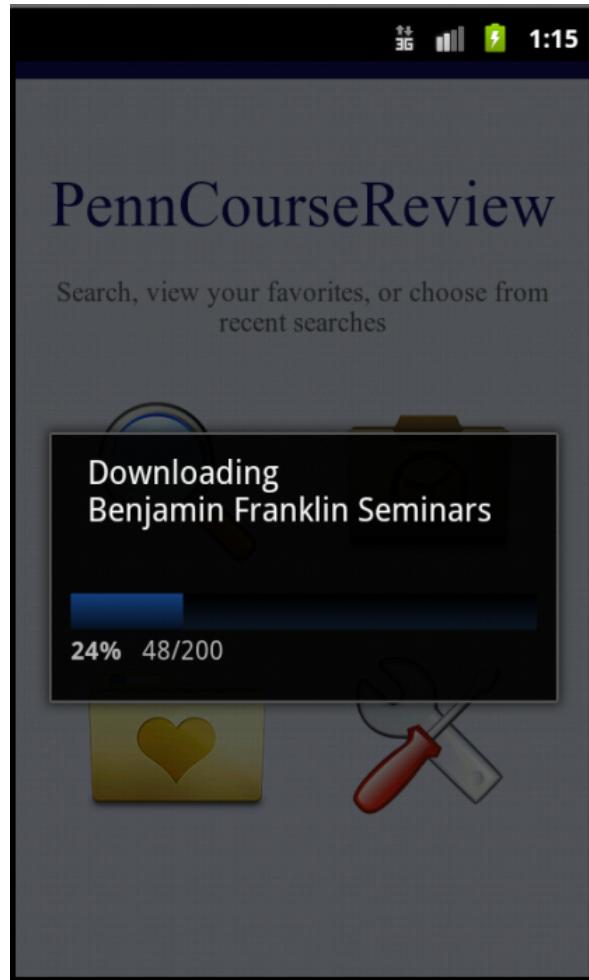


Figure 2a. Autocomplete download

2.2 Icons to Search, Recent, Favorites, Settings

The Penn Course Review start page displays the application name, brief instructions and 4 icons for the user to choose from: Search, Recent, Favorites, Settings. Pressing on the Search icon (magnifying glass on the upper left) takes the user to the Search Page (see below). Pressing on the Recent icon (folder with clock on the upper right) brings up a Recent Popup (see below).

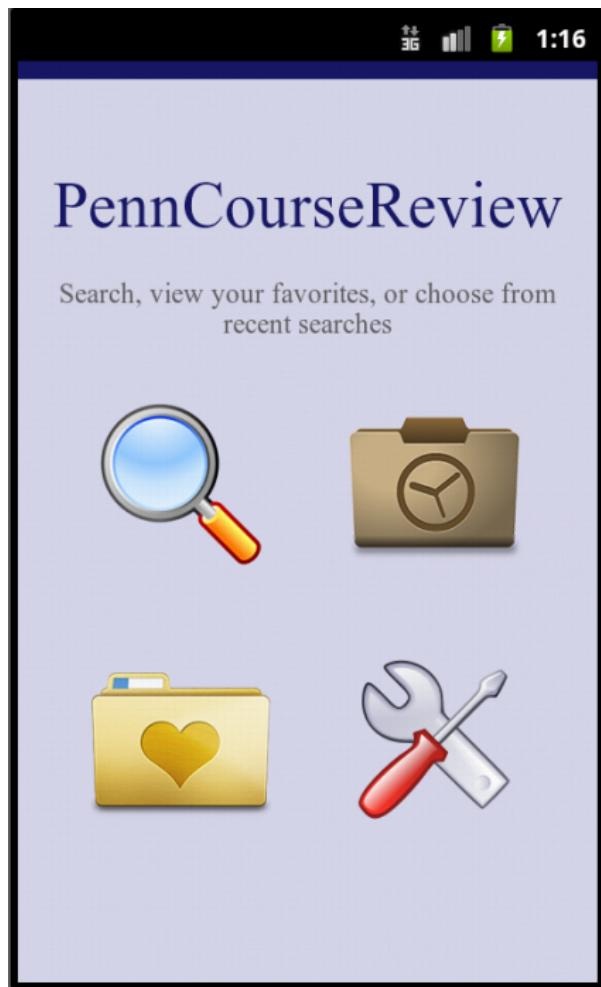


Figure 2b. Start Page

3. Search Page

The Search Page provides the user with the ability to search for reviews by course name or course ID, professor name, and department name. The user can press on the input text box and start typing the name of a course, ID of a course, name of a professor, or name of a department. Autocomplete (section 3.2) provides suggestions to autocomplete what the user has typed so far. The user can finish entering his/her search term and press enter on their phone's physical or virtual keyboard or press the Enter button. The user can Clear his/her search term if they wish to start over. The user can also directly press on the row of an autocomplete suggestion and this will automatically enter this search.

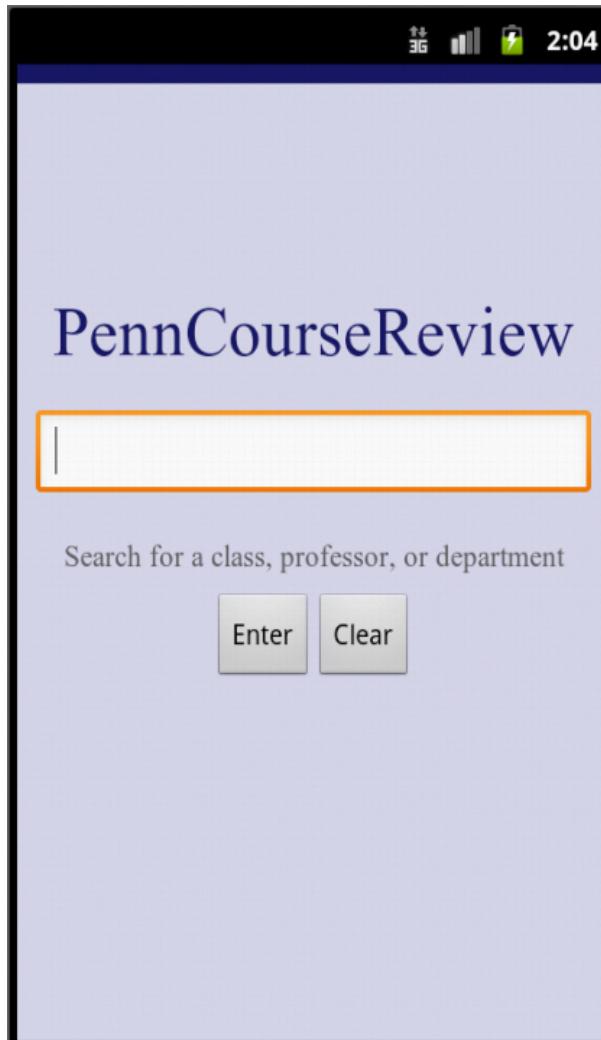


Figure 3a. Search Page

3.1 Menu items to Recent, Favorites, Settings, Quit

Pressing Menu on the user's physical Android device brings up menu options of Recent, Favorites, Settings, and Quit (Figure 3b). Pressing on the Recent tab brings up a Recent Popup over the current Search Page view (section 7). Pressing on the Favorites tab brings up a Favorites Popup (section 8). Pressing on the Settings tab takes the user to the Settings Page (section 9). Pressing on the Quit tab quits the application entirely and takes the user back to their Android home screen. After quitting the application, upon next use of the application the user is taken back to the Start Page.

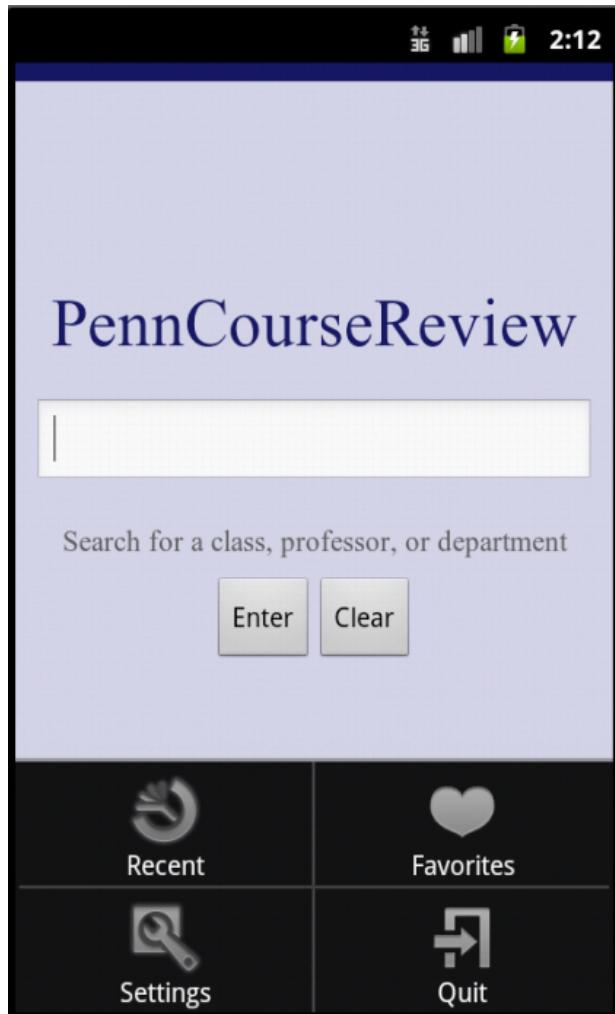


Figure 3b. Search Page menu options

3.2 Autocomplete

Autocomplete appears after a couple of characters have been typed and updates in real-time on each new character entered, providing suggestions (departments, instructors, and courses) to autocomplete what the user has typed so far. The closest match is displayed on top (see Figure 3c). The user can directly press on the row of an autocomplete suggestion and this will automatically start the search process on the item selected. As seen in Figure 3c below, the autocomplete rows are color-coded based on the type of item in the row. Courses are all a light shade of gray, instructors are shaded a pale blue, and departments are shaded a darker sky blue (seen below for CIS). This way, instructors and departments stand out when they appear as possible search queries, as there are many more courses and this search query is most common. This helps make clear to the user that he/she can search for instructors and departments as well.

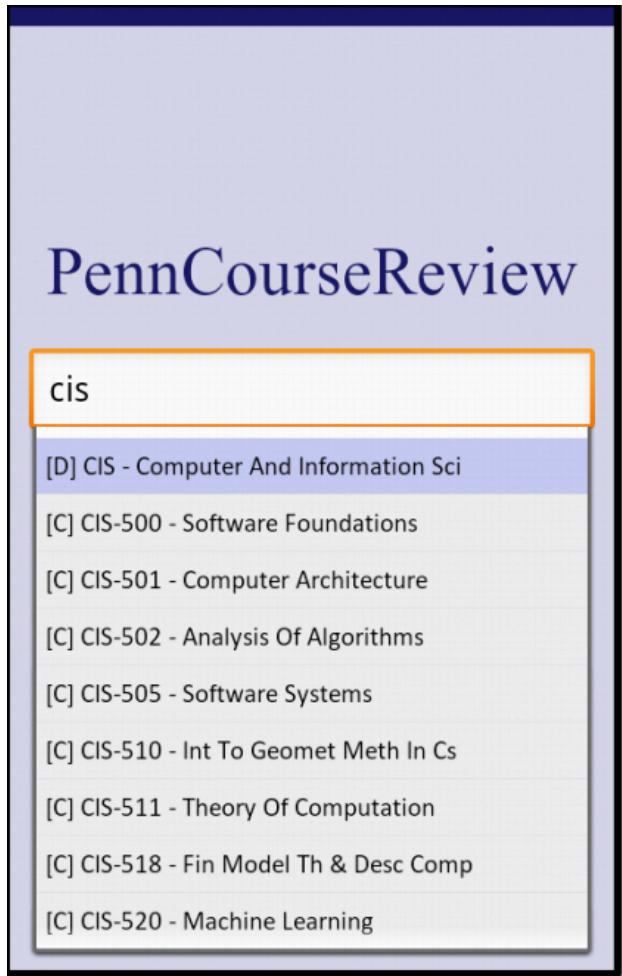


Figure 3c. Autocomplete

4. Result Page - Course

The Course Result Page shows the results upon querying a specific course. The course ID and title are listed as well as a scrollable description of the course, when available. The top right heart icon is the favorites icon (section 4.1). The lower portion shows 4 rating fields for all past offerings of the course. The 4 default rating fields are seen below (Figure 4a). The 4 fields displayed can be changed to fit the user's preferences (section 4.5). A user may press on a row to read details about a specific offering of a course (section 4.2).

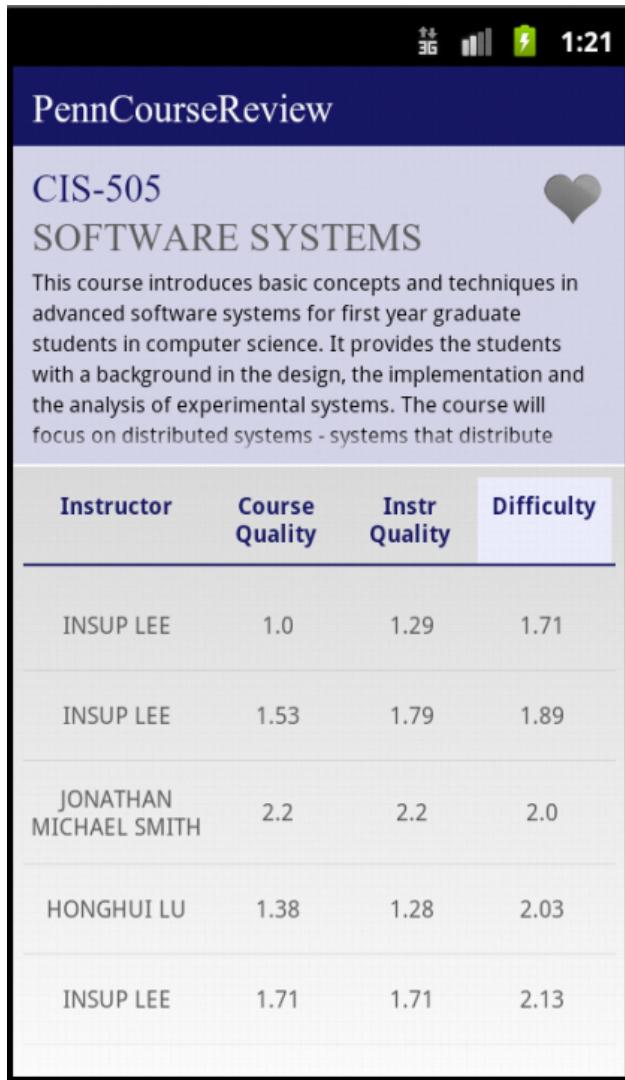


Figure 4a. Course Result Page

4.1 Favorite/Unfavorite Icon

Each user has a respective list of favorites saved on his/her phone's local database. After searching for a course, a user can favorite or unfavorite a course by pressing on the heart icon on the top right corner of the results page (see Figure 4a). Once a user has favorited a course, it is permanently saved in their favorites until it is unfavored or the user clears his/her favorites from the Settings Page (section 9). Upon favoriting a course, the upper right heart icon also shows as selected and becomes yellow (see Figure 5a for an item (instructor) that has been favorited already). This heart icon stays favorited until the user chooses to unfavorite the course by pressing on the heart icon again. A user may favorite and unfavorite an item any number of times.

4.2 Course Info Dialog

Once the course results are saved, a user can press on a row in the results list in order to get more details about a course (see Figure 4a for rows of course offerings, any of which a user may press on). After a press on a specific course, a dialog box pops up that displays all of the relevant ratings for each course (see Figure 4b). Figure 4b shows a dialog that has been scrolled partway to the comments about

the course offering. The top portion (part of it is scrolled off of the view) shows all ratings of the course, and N/A if the rating is not available for that specific course. These ratings are the same as all possible rating fields a user may select to appear in the table on a Results Page (see section 4.5, Figure 4c). Comments are also displayed if they are available for that specific course (Figure 4b shows a course offering with available comments). In order to exit the dialog, the user can press the back button to navigate back to the results page.

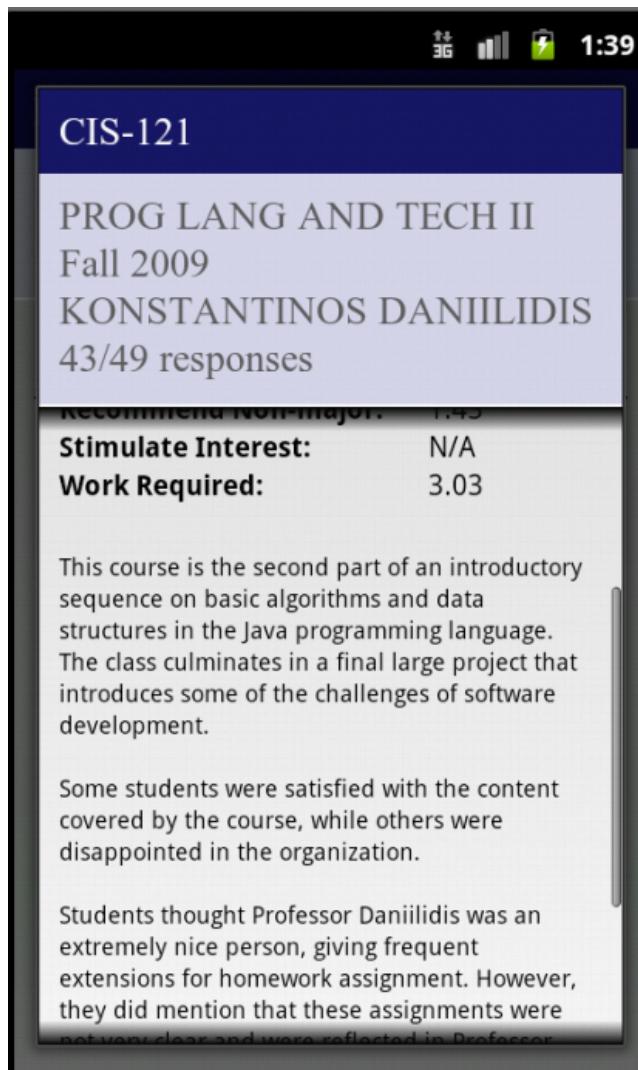


Figure 4b. Course Info Dialog

4.3 Menu items to Recent, Favorites, Settings, Quit

A Result Page has 5 menu options, as seen in Figure 5a on an Instructor Result Page (the menu options on a Course Result Page and Department Result page are the same). The menu options are the same menu options as on the Settings Page (Recent, Favorites, Settings, and Quit; see section 3.1) as well as an additional Search menu option. Pressing on the Search menu option takes the user back to the Search Page (section 3).

4.4 PennCourseReview return to Start

From any Result Page, when a user presses on the text in the top left corner that says “PennCourseReview” (see Figure 4a), he/she is taken back to the Start Page.

4.5 Rating Field Selection

On any Result Page, a user can view three different displayed ratings and a fixed fourth field that is the first, left-most field (e.g. Instructor for a Course Result Page) at a time. For courses, the three default fields are Course Quality, Instructor Quality, and Difficulty. The user can long press on a rating tab in order to choose which ratings he/she wants to view in that column. After long pressing on a tab, a dialog pops up (see Figure 4c), and the user then can select which rating he/she wants displayed on this column.

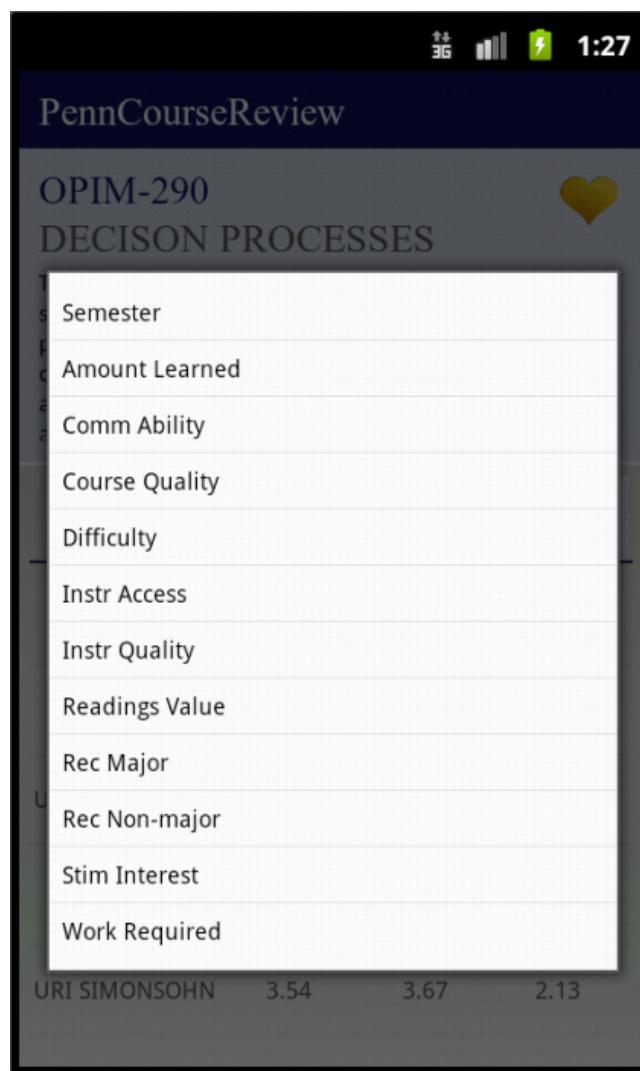


Figure 4c. Rating Field Selection

4.6 Sorting of Rating Fields

The user has the option of sorting results based off of any of the four displayed ratings for a course (including the left-most fixed field). By default, courses are sorted in order of increasing difficulty. By pressing on a tab, which is the name of the rating displayed in the column (see Figure 4a), the user can sort the results by the respective rating field. Pressing on the field name once will by default first sort

in ascending order, and pressing the same field again will then sort the column in descending order. In Figure 4a, the courses are defaulted to being sorted by increasing difficulty, and it can be seen that the difficulty column header is highlighted to show this is the field currently being sorted by. Sorting an instructor name sorts alphabetically by first name in ascending or descending order, and sorting by semester sorts chronologically by year and then by semester (Spring, Summer, Fall) within a year, in ascending or descending order. Sorting all other fields sorts the fields numerically in ascending or descending order.

5. Result Page - Instructor

The Instructor Result Page shows the results upon querying a specific instructor. The instructor name is listed. The top right heart icon is the favorites icon (section 5.1). The lower portion shows 4 rating fields for all course offerings taught by the instructor. All course offerings includes all times an instructor may have taught the same course. The 4 default rating fields are seen below (Figure 5a). The 4 fields displayed can be changed to fit the user's preferences (section 5.5). A user may press on a row to read details about a specific course taught by the instructor (section 5.2).

The screenshot shows a mobile application interface titled "PennCourseReview". At the top, the name "JEREMY SIEGEL" is displayed next to a yellow heart icon. Below this is a table with five rows, each representing a course offering. The columns are labeled "Course ID", "Semester", "Instr Quality", and "Difficulty". The data in the table is as follows:

Course ID	Semester	Instr Quality	Difficulty
FNCE-101	Spring 2002	3.7	2.8
FNCE-101	Spring 2003	3.91	2.45
FNCE-101	Spring 2004	3.86	3.1
FNCE-101	Spring 2003	3.86	2.59
FNCE-101	Spring 2002	3.67	2.5

At the bottom, there are two tabs: "Recent" and "Favorites". The "Recent" tab is selected, showing a list of recent searches. The "Favorites" tab shows a list of favorite courses. Below the tabs are three menu options: "Search", "Settings", and "Quit".

Figure 5a. Instructor Page with Favorites icon selected and menu options seen

5.1 Favorite/Unfavorite Icon

The user can favorite/unfavorite an instructor. Refer to section 4.1.

5.2 Instructor Info Dialog

The user can see more details about a specific offering from an instructor by selecting a row which corresponds to a course offering taught by the instructor. Refer to section 4.2 for info dialogs on courses.

5.3 Menu items to Recent, Favorites, Settings, Quit

Refer to section 4.3.

5.4 PennCourseReview return to Start

Refer to section 4.4.

5.5 Rating Field Selection

Refer to section 4.5.

5.6 Sorting of Rating Fields

Refer to section 4.6.

6. Result Page - Department

The Department Result Page shows the results upon querying a specific department. The department ID and name are listed on the top of the results page. The top right heart icon is the favorites icon (section 6.1). The lower portion shows 4 rating fields for all past offerings of the course. The 4 default rating fields are seen below (Figure 6a). The 4 fields displayed can be changed to fit the user's preferences (section 6.5). A user may press on a row to read details about a specific offering of a course (section 6.2). The result page shows all courses offered in the department. All ratings listed for a specific course are averaged over all past course offerings of the specific course.

The screenshot shows a mobile application interface for 'PennCourseReview'. At the top, there's a dark blue header bar with the text 'PennCourseReview' in white. Above the header, the phone's status bar shows signal strength, battery level, and the time '1:23'. Below the header, the main content area has a light purple background. The title 'CIS' is displayed in large, bold, black letters, followed by 'Computer And Information Sci' in a smaller, regular black font. To the right of the title is a gray heart icon. A note below the title states: '*Ratings listed here are averaged over all past course offerings in the department.' A table follows, listing courses with their IDs, average course quality, average instructor quality, and average difficulty. The table has a light gray header row with blue text for column headers: 'Course ID', 'Course Quality', 'Instr Quality', and 'Difficulty'. The data rows are as follows:

Course ID	Course Quality	Instr Quality	Difficulty
CIS-500	2.82	3.31	3.16
CIS-501	2.78	2.95	2.97
CIS-502	2.98	3.04	3.46
CIS-505	2.08	2.19	2.36
CIS-510	3.04	3.25	2.58
CIS-511	2.82	3.19	3.17
CIS-518	3.07	3.38	3.26

Figure 6a. Department Page

6.1 Department retrieval loading bar

As department searches take more time than course and instructor searches, the application will display a loading bar (see Figure 6b) if a user searches for a department that has not already been searched for. As the user waits for the department page to load, he/she will encounter a loading bar that displays the percentage of courses and the fraction of courses in a department that has had its data loaded into the local database. Because all query results are cached, any queries of a department after the first time querying a department will skip the loading bar and will immediately take the user to the results page.

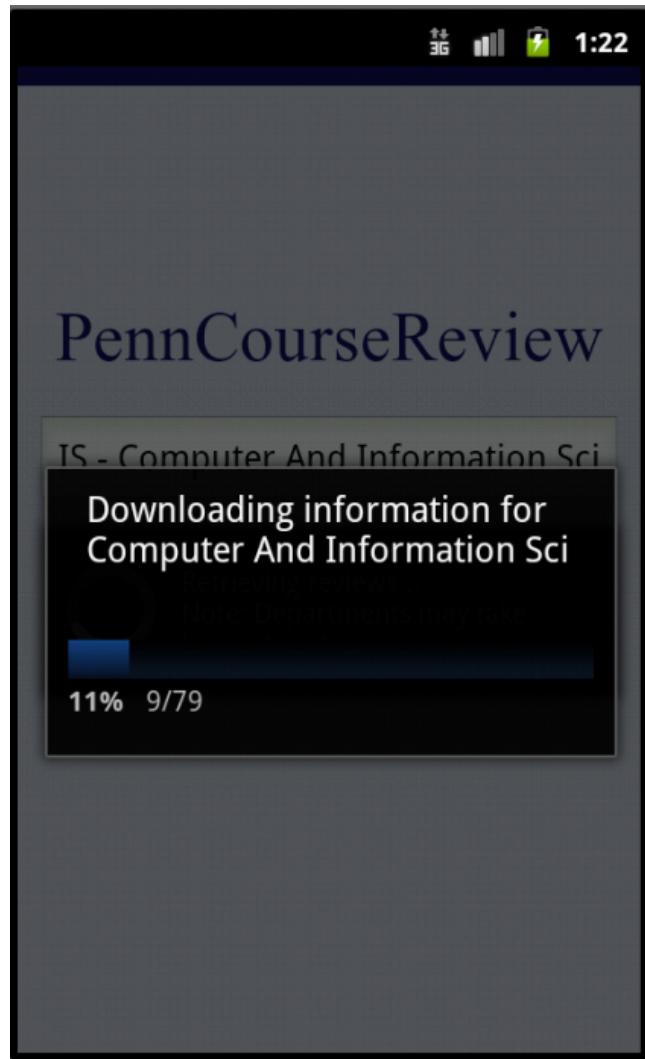


Figure 6b. Department retrieval loading bar

6.2 Favorite/Unfavorite Icon

A user can favorite/unfavorite a department. Refer to section 4.1.

6.3 Menu items to Recent, Favorites, Settings, Quit

Refer to section 4.3.

6.4 PennCourseReview return to Start

Refer to section 4.4.

6.5 Rating Field Selection

Refer to section 4.5.

6.6 Sorting of Rating Fields

Refer to section 4.6.

6.7 Course Selection

A user can press on a row which corresponds to a course in the department, and the user will automatically be taken to the Course Result page for that specific course (see section 4).

7. Recent Popup

From the search page, a user can press Menu on his/her Android device to access recent searches from the menu options. By clicking on the Recent tab, a dialog pops up with a user's most recent searches (see Figure 7a). The searches are listed with the most recent search appearing on top and the least recent search on the bottom. Until the user clears his/her History from the Settings page, all past searches are kept in the Recent searches list. From the Recent dialog, the user may press on any of the items and this will take him/her directly to the Result page for that item. Because these are all searches that have already been executed in the past, all results are cached (assuming the user did not clear his/her cache from the Settings page) and so all results can be displayed immediately.

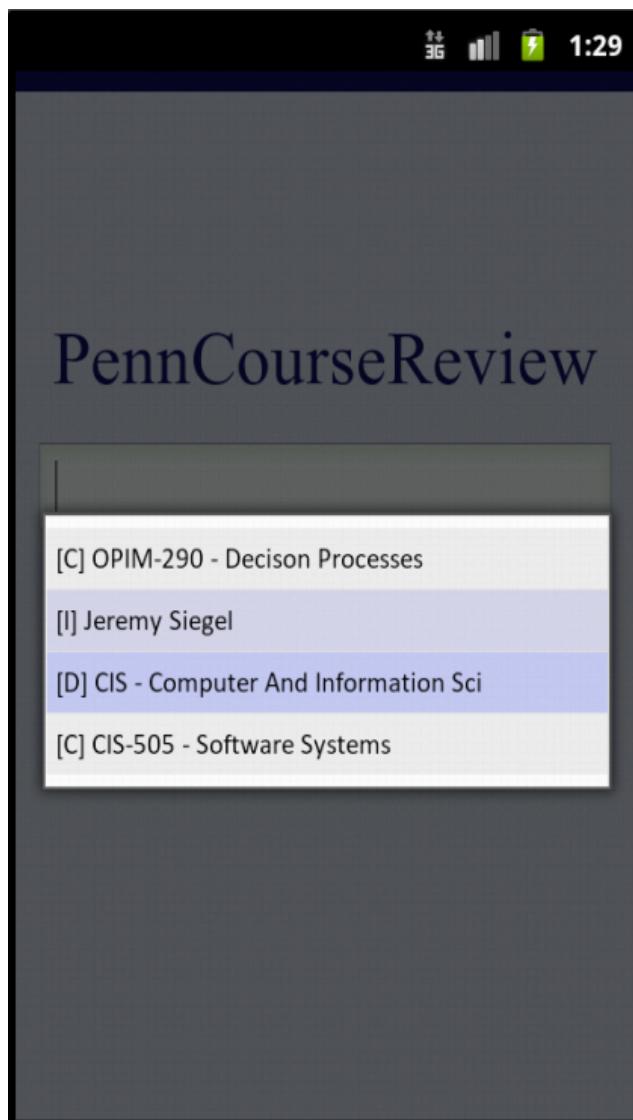


Figure 7a. Recent Popup

7.1 Color coded options

The rows in the Recent Popup are color-coded based on the type of item in the row. Courses are all a light shade of gray, instructors are shaded a pale blue, and departments are shaded a darker sky blue (seen below for CIS). See section 3.2 for a more detailed discussion on the color coding, which is the same color coding scheme used in displaying Autocomplete suggestions.

8. Favorites Popup

From the search page, a user can press Menu on his/her Android device to access Favorites from the menu options. By clicking on the Favorites tab, a dialog pops up with a user's Favorites (see Figure 7a, same style as Recent Popup). The favorites are listed in the order the user favorited them, with the most recent favorite on top and the least recent on the bottom. Until the user clears his/her Favorites list from the Settings page, all favorited items are kept in the Favorites list. From the Favorites dialog, the user may press on any of the items and this will take him/her directly to the Result page for that item. Because these are all searches that have already been executed in the past, all results are cached (assuming the user did not clear his/her cache from the Settings page) and so all results can be displayed immediately.

8.1 Color coded options

See section 7.1.

9. Settings Page

The user can reach the Settings Page from the icon on the Start Page or from the Settings menu option on the Search or any Result Page. The Settings Page provides the user with settings information and options.

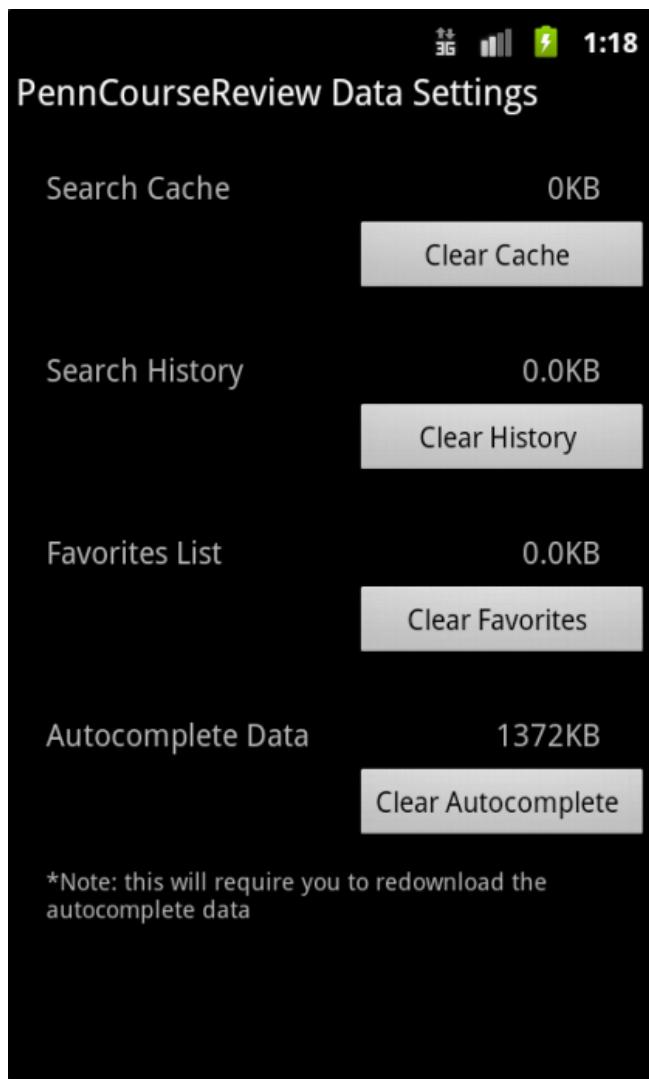


Figure 9a. Settings Page

9.1 Cache sizes

The size of the search cache, search history, favorites list, and autocomplete databases are all listed (Figure 9a).

9.2 Clear options

The buttons with options to clear certain portions of data can be seen in Figure 9a. Pressing on the top button, Clear Cache, clears the search cache of all cached queries the user has previously executed. Future searches will have to be retrieved again from the database and are not locally cached (all query results are cached to speed up future result retrieval times for any query that has already been executed). Pressing on the second button, Clear History, clears the user's locally saved history of recent searches. Pressing on the History icon from the Start Page or History menu option from the Search or Result Pages immediately after clearing the search history will show no recent history. Pressing on the third button, Clear Favorites, clears the user's locally saved list of favorites. This also unselects the yellow heart icon for each item that was previously favorited. Pressing on the Favorites icon from the Start Page or Favorites menu option from the Search or Result Pages immediately after clearing the favorites list will

show no favorites. Pressing on the last button, Clear Autocomplete, will completely clear the user's autocomplete data stored on his/her phone. This will require the user to redownload the autocomplete data, as this is required for autocomplete to work as a user types in a search term. We require the user to have autocomplete data to access the search page at all, so upon clearing autocomplete, when the user pushes back to leave the Settings page, the user will automatically be taken to the Start Page with autocomplete downloading (section 2.1). This option is here for any user that may have incomplete autocomplete or some corrupted version of the autocomplete data. This may happen from rare cases where the user's wifi or signal disconnected in the middle of autocomplete download, and the autocomplete download failed to resume upon a signal returning.

Technical Documentation

1. Package/Class Hierarchy

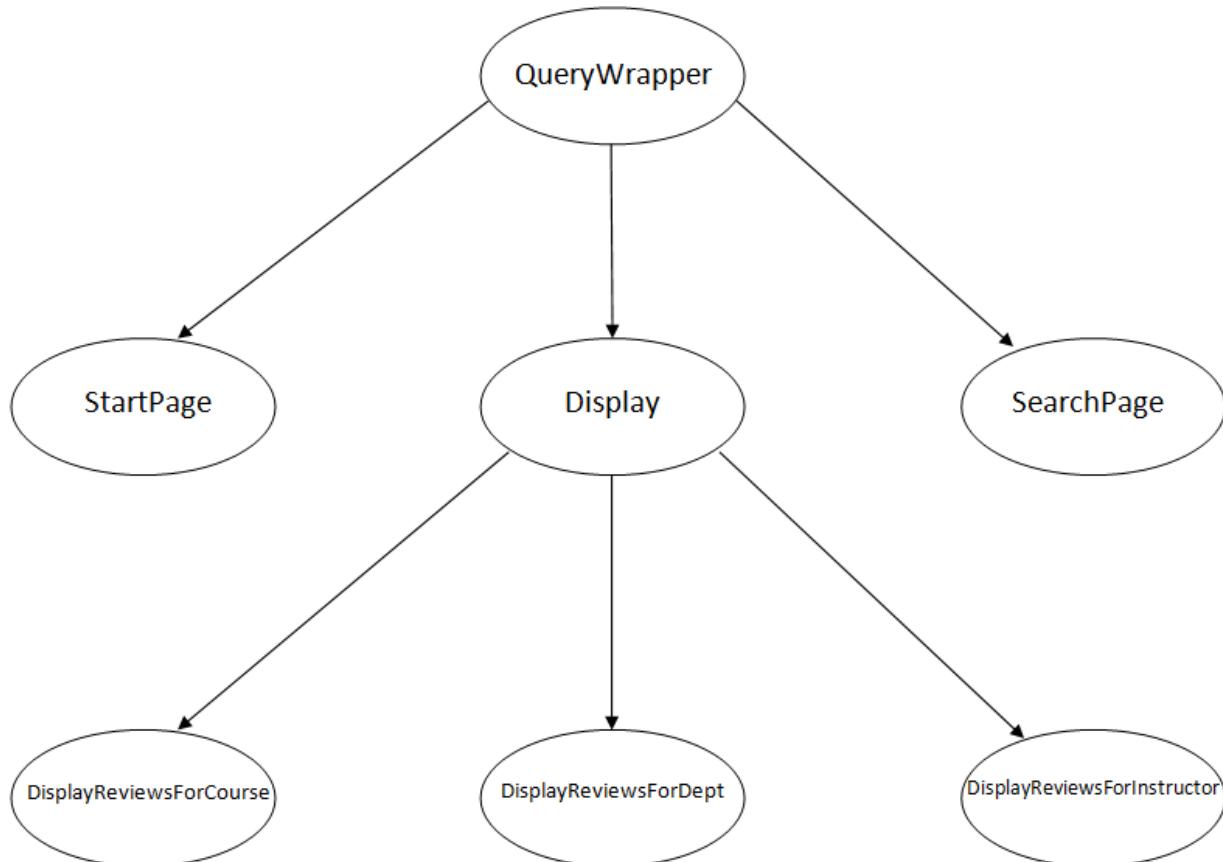
1.1 Overview

Our primary packages are divided as follows:

- o edu.upenn.cis.cis350.backend - this contains most of our helper classes, such as query classes, JSON parsing classes, normalizing classes, etc.
- o edu.upenn.cis.cis350.database - this contains all of our SQLiteDatabase related files which manages the local database the application uses on the Android device.
- o edu.upenn.cis.cis350.display - this contains all of our Android Activity classes that basically control the user interface and interaction for the application.
- o edu.upenn.cis.cis350.objects - this contains all of our Java object wrapper classes.

Our test project package is essentially identical except that the packages are named “edu.upenn.cis.cis350.test.[specific package]”.

1.2 Activity Hierarchy



The above diagram shows the class hierarchy for the Activity classes. AuthenticationPage and SettingsPage are omitted because they are standalone Activities. QueryWrapper is the main parent class

for most of the Activities we use because it contains many of the functionalities that are common for StartPage, SearchPage, and all of the result pages, such as Dialog creation and Asynchronous Tasks for querying local database or PCR server. Display is then the parent class for all of the result Activities as it contains common code that are relevant for all of the result Activities, such as formatting code, long press listener for columns, dialog creation, etc.

2. Class Descriptions

2.1 `edu.upenn.cis.cis350.display`

2.1.1 `AuthenticationPage.java`

Figure 1a – Authentication Page

Relevant Activities and Classes:

`AuthenticationPage` (activity), `Constants`, `EasySSLSocketFactory`, `EasyX509TrustManager`, `AuthCache`, `FpsPennGroupsHasMember`

Overview of Classes:

The AuthenticationPage (activity) is the activity corresponding to Figure 1a. From there, the user can retrieve his or her serial number online (not affiliated with the mobile application). Once the user inputs his serial number in the text field, the application will execute an HTTP Get request to retrieve the associated Penn Key that is associated with the given serial number. To accomplish this, the Apache-written classes EasySSLSocketFactory, EasyX509TrustManager are used in order to have our application trust all external websites without certification. The AuthCache class is used so the user doesn't have authenticate himself everytime he or she uses the application - their serial number is stored in the cache and is cleared every 30 days. FpsPennGroupsHasMember is the class that is used to verify that the Penn Key retrieved is actually a valid, existing student. Once the user has been authenticated, he or she is taken to the Start Page.

2.1.2 `QueryWrapper.java`

Superclass, no associated figure

Relevant Activities and Classes:

`QueryWrapper.java` (activity), `Constants`, `Normalizer`, `Parser`, `AutoCompleteDB`, `CourseSearchCache`, `DepartmentSearchCache`, `RecentSearches`, `Course`, `CourseAverage`, `Department`, `KeywordMap`

Overview of Classes:

No actual application page is associated with this class as this class is only used to duplicate functionalities in various sub-pages. The primary use of this class is to show the same Dialogs on all of the subclasses as well as having the keyword processing functionality. This means all of the subclasses can take a given keyword and first search the local database and then query the PCR server to get the necessary information to populate the results page.

2.1.3 `StartPage.java`

Figure 2a – Start Page

Relevant Activities and Classes:

`StartPage` (activity), `QueryWrapper` (superclass), `AutoComplete`, `Constants`, `KeywordMap`

Overview of Classes:

The StartPage (activity) is the activity corresponding to Figure 2a. Upon arriving at the StartPage for the first time, the application first downloads the autocomplete data used in the search (this takes roughly 1 minute). The AutoComplete class is used to retrieve and parse the data from the penn apps API. The user can choose either to go to the Search Page, Favorites, Recent, or Settings. KeywordMap is simply an object-data class used to store the data retrieved from the online API.

2.1.4 SearchPage.java

Figure 3a – Search Page

Relevant Activities and Classes:

SearchPage (activity), **QueryWrapper** (superclass), **Constants**, **QueryWrapper**

Overview of Classes:

The SearchPage (activity) is the activity corresponding to Figure 3a. From this point on, we would like to point out that the user can press “Menu” on their phone and a built-in options menu will popup from the bottom, where the user can either choose to view Recent searches, Favorite searches, go to the Settings page, or quit. Recent and Favorite searches popup as Dialogs. If the user wants to search for a course, instructor or department, the autocomplete list will automatically drop down once the user starts typing. The list is also color coded by course, instructor, or department.

2.1.4 Display.java

Superclass, no associated figure

Relevant Activities and Classes:

Display.java (activity), **QueryWrapper** (superclass), **Constants**, **Sorter**, **Course**, **CourseAverage**, **KeywordMap**, **Ratings**

Overview of Classes:

No actual application page is associated with this class as this class is only used to duplicate functionalities in various sub-pages. This page is used mainly to contain the code used to populate the various rows and columns of the results page as well as providing the same sort and changing column functionalities.

2.1.5 DisplayReviewsForCourses.java

Figure 4a – Course Page

Relevant Activities and Classes:

DisplayReviewsForCourse (activity), **Display** (superclass), **Constants**, **CourseSearchCache**, **KeywordMap**, **Parser**, **Sorter**

Overview of Classes:

DisplayReviewsForCourse (activity) is the activity corresponding to Figure 4a. This is the page where all results for a given course are displayed. If the user has searched for a given course number before, then it first goes to the cache (CourseSearchCache) to check, and returns the results. If not, then the CourseSearchCache calls Parser, which is the class that retrieves data from the online API, then inserts it into the cache. On the course results page, the user can sort by ratings, semester, or course name - all of the sorting is done in the backend in the Sorter class. The user can also long press on any of the column fields (except for course name), and a Dialog will popup in which the user can choose what other ratings to view. Finally, the user can press on a row in the results list and another dialog will popup that shows a detailed description of the course ratings and information.

2.1.6 DisplayReviewsForInstructor.java

Figure 5a – Instructor Page

Relevant Activities and Classes:

DisplayReviewsForInstructor (activity), **Display** (superclass), **Constants**, **CourseSearchCache**, **KeywordMap**

Overview of Classes:

DisplayReviewForInstructor (activity) is the activity corresponding to Figure 5a. It is very similar to **DisplayReviewForCourse**, but instead displays reviews for instructors.

2.1.7 DisplayReviewsForDept.java

Figure 6a – Department Page

Relevant Activities and Classes:

DisplayReviewsForDept (activity), **Display** (superclass), **Constants**, **DepartmentSearchCache**, **Department**, **KeywordMap**, **CourseAverage**, **Parser**

Overview of Classes:

DisplayReviewsForDept (activity) is the activity corresponding to Figure 6a. It is very similar to the other **DisplayReviewForXXXX** classes, but instead displays reviews for departments. If the **DepartmentSearchCache** cannot find a given department in the cache, then it calls **Parser** to retrieve the data from the online API. Similar to the autocomplete, a progress bar will appear that updates the user on all the data the application is retrieving in real time, since **Department** searches are slower. Once all the results are retrieved, the **Parser** takes an average over all courses in a department to end up with an average rating for each field per course - these ratings are stored in a class called **CourseAverage**.

2.1.8 SettingsPage.java

Figure 9a – Settings Page

Relevant Activities and Classes:

SettingsPage (activity), **Constants**, **AutoComplete**, **CourseSearchCache**, **DepartmentSearchCache**, **RecentSearches**

Overview of Classes:

The **SettingsPage**(activity) is the activity corresponding to Figure 9a. On this page the user can view the database size the application is currently using on the Android device for each relevant database, such as local search cache, recently searched list, favorite list, and autocomplete data. The user can also choose to clear individual database to conserve phone storage space.

2.2 edu.upenn.cis.cis350.backend

2.2.1 JSONRequest.java

Tester: **JSONRequestTest.java**

Overview:

JSONRequest class simply contains one static method that is used by many other classes to retrieve data from the online API. It is given a URL and returns the **JSONObject** that is on that URL.

2.2.2 Normalizer.java

Tester: **NormalizerTest.java**

Overview:

The **Normalizer** class contains two static methods that we use to manipulate Strings. One method normalizes keywords so that they match with the keywords in the **AutocompleteTable** and can be inserted into the cache database. The other method takes a given String and capitalizes the first letter of every

word (used to display course/instructor/department names properly for autocomplete).

2.2.3 Parser.java

Tester: **ParserTest.java**

Overview:

Parser is the class that is used to retrieve and filter various data from the online API. For example, there are methods to retrieve all data associated for a given course, for a given instructor, and for a given department. The data is stored into one of the .object classes (Course, Rating, Department, Instructor).

2.2.4 Sorter.java

Tester: **SorterTest.java**

Overview:

Sorter is the class on the backend that is used to sort all results by a given rating field, by a semester, or by course/instructor name.

2.2.5 AutoComplete.java

Overview:

AutoComplete class contains three methods that are used to retrieve all the data associated with autocomplete from the online API. It retrieves the JSON objects for all instructors and departments and courses and returns the data organized in an array list of KeywordMaps.

2.3 edu.upenn.cis.cis350.database

2.3.1 DatabaseHelperClass.java

Overview:

This class contains the actual SQLiteOpenHelper and SQLiteDatabase objects. It also contains all of the constants for each table name and table create statements. This class is used mainly to avoid code duplication as well as having a global reset to all of the local databases through the SQLiteOpenHelper versioning system.

2.3.2 AuthCache.java

Tester: **AuthCacheTest.java**

Schema:

```
    AuthenticationTable
    (
        auth_key char(10) PRIMARY KEY NOT NULL,
        year integer NOT NULL,
        day integer NOT NULL
    )
```

Overview:

AuthCache extends DatabaseHelperClass and manages the local database associated with storing the Penn-authenticated key. Keys are reset after 30 days (a constant specified in Constants.java). When a key is available in this database, the user can skip AuthenticationPage and go straight to

StartPage on application start.

2.3.3 AutoCompleteDB.java

Tester: **AutoCompleteDBTest.java**

Schema:

```
AutoCompleteEntries
(
    path char(50) PRIMARY KEY NOT NULL,
    name char(50) NOT NULL,
    course_id char(50),
    course_id_norm char(50),
    type int NOT NULL,
    year int NOT NULL
)
```

Overview:

The AutocompleteDB extends DatabaseHelperClass and manages the local database that contains all of the relevant autocomplete terms as well as the path associated with each term. The path is the URL we use to query the PCR server to get more information for the given keyword. The type field is to distinguish if the keyword is for a course (0), instructor (1), or department (2). These values are also defined in Constants.java. This database is reset every year since we assume that the PCR server will refresh its data annually.

2.3.4 CourseSearchCache.java

Tester: **CourseSearchCacheTest.java**

Schema:

```
CourseResults
(
    p_id integer PRIMARY KEY AUTOINCREMENT,
    name char(50) NOT NULL,
    course_alias char(20) NOT NULL DEFAULT '',
    description char(700),
    semester char(50),
    course_id char(50) NOT NULL,
    comments char(300),
    instructor_id char(50) NOT NULL,
    instructor_name char(50) NOT NULL,
    instructor_path char(50) NOT NULL,
    num_reviewers integer NOT NULL DEFAULT 0,
    num_students integer NOT NULL DEFAULT 0,
    course_path char(50) NOT NULL,
    ratings_amountLearned float,
    ratings_commAbility float,
    ratings_courseQuality float,
    ratings_difficulty float,
    ratings_instructorAccess float,
    ratings_instructorQuality float,
    ratings_readingsValue float,
```

```

        ratings_recommendMajor float,
        ratings_recommendNonMajor float,
        ratings_stimulateInterest float,
        ratings_workRequired float,
        section_id char(50) NOT NULL,
        section_alias char(50) NOT NULL,
        section_path char(50) NOT NULL,
        section_name char(50) NOT NULL,
        section_number char(20) NOT NULL,
        type int NOT NULL,
        date int NOT NULL
    )

```

Overview:

The CourseSearchCache extends DatabaseHelperClass and manages the local database that saves all of the cached information for previously searched courses and instructors. Again, the type field indicates if a given row is associated with a course search or an instructor search.

2.3.5 DepartmentSearchCache.java

Tester: **DepartmentSearchCacheTest.java**

Schema:

```

DepartmentResults
(
    d_id integer PRIMARY KEY AUTOINCREMENT,
    dept_name char(50) NOT NULL,
    dept_id char(20) NOT NULL,
    dept_path char(50),
    course_name char(50) NOT NULL,
    course_id char(20) NOT NULL,
    course_path char(50) NOT NULL,
    ratings_amountLearned float,
    ratings_commAbility float,
    ratings_courseQuality float,
    ratings_difficulty float,
    ratings_instructorAccess float,
    ratings_instructorQuality float,
    ratings_readingsValue float,
    ratings_recommendMajor float,
    ratings_recommendNonMajor float,
    ratings_stimulateInterest float,
    ratings_workRequired float,
    date int NOT NULL
)

```

Overview:

The DepartmentSearchCache extends DatabaseHelperClass and manages the local database that saves all of the cached information for previously searched departments.

2.3.6 RecentSearches.java

Tester: **RecentSearchesTest.java**

Schema:

```
RecentSearches/Favorites
(
    s_id integer PRIMARY KEY,
    keyword char(50) NOT NULL
)
```

Overview:

The RecentSearches extends DatabaseHelperClass and manages two tables - RecentSearches and Favorites. Both tables simply store a primary key and the keyword for the search/favorite term.