**Software Engineering CSC4350**

**Spring 2016**

**Automated Student Calendar**

**Project Team 3**

**Final Document**

4/21/2016

**De’jon Miller** – Team coordinator /Programmer (UI implementation)

**Todor Guichin** – Programming architect (OCR)

**James Jackson** – Back-up programmer

**Alex Shyu** – Programmer

**Joseph Yun** –Documentation, Back-up programmer

**Database –** File system\*

**Software Architecture –** Android (Java/XML)

**Table of Contents**

**Introduction**1

Problem statements

**Requirements Elicitation & RTM with “Shall” statement**2, 3

RTM

Shall statements

**Object Design**4

Functional Point Cost (FPC)

Category Interaction Diagram

## **Test Cases**5, 6, 7, 8

## **Test cases**

## COCOMO

## **Rationale for entire project design & implementation**9, 10

Use-case rationales

Fundamental rationales

**Problem statement**

**Why automated student calendar? (Rationale)**

* No current product with this feature
* Inconveniency for user to input important dates into calendar
* Gather concise information from syllabus
* Avoid irrelevant information
* Centralized accessibility for multiple syllabi
* Easier to look up textbook
* Integration of OCR into calendar / Utilization of open-source API
* Convenience of keeping track of professor contact information

**What’s the problem?**

Students receive multiple syllabi with extraneous information, so they need a convenient way to keep relevant information in one place.

**Who is it used for?**

Mainly used for college students

**Problem statement (Functional requirements)**

**Lazy student Calendar (LSC)** is an automated method that uses student syllabus to collect important dates, contact information, and textbook ISBN (if available). **LSC** uses phone camera and OCR\* API\* in order to scan images for relevant information for the students to use. Students will be able to create an account in **LSC** that stores their username/password and holds unique account information in orderto look up book prices, contact professors, and record important dates to become more successful student.

**LSC** will allow addition and edit of dates in native mobile calendar upon image capture or at a time of students choosing. **LSC** will contact professor/TA upon image capture or at a time of students choosing. **LSC** will search for textbook prices upon image capture or at a time of students choosing. Textbook search will utilize native mobile browser. **LSC** will assume that students’ phone meet minimum SDK\* requirement.

**LSC** will store the data extracted from images in internal phone memory using a database (still deciding which database we will use). Students will be able to update **LSC** via Google Play Store.

Pg. 1

**RTM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entry** | **Paragraph** | **Shall** | **Type** | **UseCaseName** | **Implemented (Y/N)** |
| 1 | 1 | **LSC** shall automatically collect important dates, contact information, and textbook ISBN | SW | Use case #1 | Y |
| 2 | 1 | **LSC** shall use phone camera and OCR\* API\* to scan images for relevant information. | SW, HW | Use case #1 | Y |
| 3 | 1 | **LSC** shall allow user (mainly students) to create username/ password which will hold unique account information. | SW, SWC | (Login system) | N |
| 4 | 1 | **LSC** shall look up book prices based on ISBN found in the syllabus. | SW, SWC | Use case #4 | Y |
| 5 | 2 | **LSC** shall allow user to input dates manually through native calendar application. | SW, SWC | Use case #3 | Y |
| 6 | 2 | **LSC** shall allow user to look up book prices at a later time upon users’ choosing. | SW | Use case #5 | Y |
| 7 | 2 | **LSC** shall contact professor/ TA upon image capture or at users’ choosing time | SW | (Not yet determined)  “Nice to have” | N |
| 8 | 2 | **LSC** shall use native mobile browser for textbook search. (prices) | SW | Use case #4 | Y |
| 9 | 2 | **LSC** will assume that students’ phone meet minimum SDK\* requirement. | SWC | (System requirement)  No use case used | Y |
| 10 | 2 | **LSC** shall store the data extracted from images in internal phone memory using a file system. | SW | Use case #1 | Y |
| 11 | 2 | **LSC** shall use “Google Play store” to update its version | SW | (System requirement)  No use case used | Y |

Pg. 2

**“Shall” statements**

* **LSC** shall automatically collect important dates, contact information, and textbook ISBN (if it’s available).
* **LSC** shall use phone camera and OCR\* API\* to scan images for relevant information.
* **LSC** shall allow user (mainly students) to create username/ password which will hold unique account information.
* **LSC** shall look up book prices based on ISBN found in the syllabus.
* **LSC** shall allow user to input dates manually through native calendar application.
* **LSC** shall allow user to look up book prices at a later time upon users’ choosing.
* **LSC** shall contact professor/ TA upon image capture or at users’ choosing time.
* **LSC** shall use native mobile browser for textbook search. (prices)
* **LSC** will assume that students’ phone meet minimum SDK\* requirement.
* **LSC** shall store the data extracted from images in internal phone memory using a database (undecided).
* **LSC** shall use “Google Play store” to update its version.

Pg. 3

**Object Design**

**UFP (Weighting Factor Estimate)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Weighing Factor Estimate** | | | | | | |
| **Measurement Parameters** | | | | | | |
|  | **Count** |  | Value | Category |  |  |
| **Number of User Inputs** | 1 | x | 3 | Simple |  | 6 |
| **Number of User Outputs** | 4 | x | 5 | Average |  | 20 |
| **Number of User Inquires** | 1 | x | 4 | Average |  | 4 |
| **Number of Internal Files** | 1 | x | 10 | Average |  | 10 |
| **Number of External Interface of files** | 3 | x | 7 | Average |  | 21 |
| **Grand total (FP)** | | | | | | 61 |

**Rating Estimate of Categories (VAF)**

|  |  |  |
| --- | --- | --- |
|  | Category | Rating |
| 1 | Does the system require reliable backup and recovery | 0 |
| 2 | Are data communications required | 0 |
| 3 | Are there distributed processing functions | 0 |
| 4 | Is performance critical | 4 |
| 5 | Will the system run in existing, heavily utilized operational environment | 3 |
| 6 | Does the system require online data entry | 0 |
| 7 | Multiple Screens or Operations | 5 |
| 8 | Are the master files updated online | 0 |
| 9 | Are the input, output, inquires complex | 2 |
| 10 | Is the internal processing complex | 4 |
| 11 | Is the code designed to be reusable | 5 |
| 12 | Are conversions and installations included in the design | 2 |
| 13 | Is the system designed for multiple installations for different organizations | 4 |
| 14 | Is the application designed to facilitate change and be ease of use to the user. | 5 |
|  | Total sum of all category ratings | 34 |

pg. 4

**Value Adjusted Factor Calculation (VAF)**

VAF = [0.65 + 0.01 \* (sum of all category ratings)]

0.65 + 0.34 = 0.99

**Functional Point Cost (FPC)**

FPC = UFP \* VAF

61 \* 0.99 = **60.39**

Pg. 5

**Test cases**

|  |  |
| --- | --- |
| ***Test-case identifier*** | Image Capture (OCR) |
| ***Test location*** | Samsung Galaxy S4 (v4.4.2 Kit-Kat) |
| ***Input*** | Captured image through device: Syllabus format |
| ***Oracle*** | Expected result: Fast recognition of each word with valid characters.  Actual result: Processing speed is depended on the size of the input. The result returns slower than expected, especially when the input was very large; Nonetheless, recognition will process in acceptable time constraint. Accuracy falls within acceptable constraint. |
| ***log*** | Format of the output contains one string, separated by spaces. Accuracy still remains our major focus in development. |

|  |  |
| --- | --- |
| ***Test-case identifier*** | Book price search (Automatic) |
| ***Test location*** | Samsung Galaxy Note 2 (v4.4.2 Kit-Kat) |
| ***Input*** | ISBN captured by OCR |
| ***Oracle*** | Expected result: With the ISBN number given by the OCR, this function should open browser application to look up textbook information online.  Actual result: correctly opens up browser as it should. No issues found. |
| ***log*** | Output follows desired-output format. |

|  |  |
| --- | --- |
| ***Test-case identifier*** | Book price search (Manual) |
| ***Test location*** | Samsung Galaxy S5 (v5.0.1 Lollipop) |
| ***Input*** | User selected information (ISBN number) from history |
| ***Oracle*** | Expected result: With the ISBN number provided by the history, this function should open browser application to look up textbook information online.  Actual result: correctly opens up browser with containing information. No issues found. |
| ***log*** |  |

|  |  |
| --- | --- |
| ***Test-case identifier*** | Calendar Management (Automatic) |
| ***Test location*** | Samsung Galaxy S4 (v4.4.2 Kit-Kat) |
| ***Input*** | Valid dates from OCR |
| ***Oracle*** | Expected result: Automatically populating dates in android calendar application with its contents.  Actual result: Accuracy and speed depends on how accurate OCR processes taken picture. Although accuracy falls below desired constraints, however, with accurate input & OCR recognition, result is valid and accurate. |
| ***log*** | Dates with valid input follows desired-output request. Partial results (invalid OCR recognition) does not return valid result. |

Pg. 6

|  |  |
| --- | --- |
| ***Test-case identifier*** | Calendar Management (Manual) |
| ***Test location*** | Samsung Galaxy S5 (v5.0.1 Lollipop) |
| ***Input*** | User selected information (ISBN number) from history |
| ***Oracle*** | Expected result: Upon selection from the history & confirmation by the user, function should populate corresponding dates and its event to calendar application.  Actual result: Although results were similar to expected result, time and the accuracy still remains the issue of this functionality. |
| ***log*** | Dates with valid input follows desired-output constraints. Potential problem (time & accuracy) expected. |

Pg. 7

**COCOMO**

**Project Complexity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software Project** | **ab** | **bb** | **cb** | **db** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |

**LOC**

|  |  |
| --- | --- |
| **Programming Language** | **LOC/FP (average)** |
| Object-Oriented Languages | 30 |

K = 2.48

Effort (E) = ab (KLOC)b b = 5.75

Duration (D) = 4.61

Source: http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/gamel/cocomo.html

Pg. 8

Use Case Rationale

**UC\_001\_Image\_Capture\_OCR:**

This part of the application is required because there are no current available applications with such a feature out there to help students be more organized and still be lazy. This is the most critical use case to the program, as all the data processed in later use cases is captured in this step. The OCR will allow the student to automatically capture syllabi information and enter it into their calendar.

**UC\_002\_Calendar\_Management\_Automatic:**

The automatic calendar entry feature is necessary for the “automation” component of Lazy Student Calendar. This use case gives the user the option to automatically import captured data from syllabi and (in one step) import the information to their native android calendar.

**UC\_003\_Calender\_Management\_Data\_History**

This feature is a nice-to-have function for the user to have the option of manual calendar entry either at point of data capture or at a later time. If the user is unable or unwilling to import data into their calendar at time of capture, this use case allows storage of syllabi calendar data into their account for later retrieval and calendar import.

**UC\_004\_Book\_Price\_Search\_Automatic**

This is a critical system function carrying with the theme of automation, involving automatic (optional) textbook price search from captured data. As part of the core functionality, this feature uses data from the ISBN portion of listed textbooks in student syllabi and gives the option at time of capture to search for books online. The search is carried out via a native mobile browser launch into a Google Shopping ISBN search.

**UC\_005\_Book\_Price\_Search\_Manual\_History**

This feature is a nice-to-have system operation to give the user freedom to search book prices at a later date. This is an additional feature that makes use of stored data for the user in the event that the user chooses not to search for textbooks at the time of syllabus data capture.

Pg. 9

**Fundamental Rationales**

**Android platform**

One of the biggest reason why we have decided to go with Android platform was due to the fact that Android application is using the required programming language: Java. Also the main intent of our application involves around taking pictures and the information that will be extracted within those taken pictures; therefore it was reasonable for us to work with Android platform, in which it supports all of our needs.

**Tesseract (open-source API, OCR)**

Open-source API that is widely used OCR technologies in many different platforms. In order to extract data from user-taken image, syllabi, this part of our application is one of the most essential tool that we will incorporate in our application.

**Native Applications (Calendar, Contact, E-mail, Browser)**

Native applications will mainly include Calendar, Contacts, Email, and Browsers. Due to our scope of time, we wanted to avoid putting time and effort in developing pre-existing functionality that is already built in android platform. These built-in application will allow our application to be efficient and mobile friendly.

Pg. 10