e advisor



by  
Noor Najjar  
Xiaorong Guo  
Qingwei Kong  
Sarah Lee  
Chris Shiohama  
Justin Syhongpan

Software Engineering  
Group Project: Report 1  
Group 6

**Section 1: Team Organization**

Our group organization is called E Advisor. The team members and their respective backgrounds are the following:

1. Noor Najjar: Project Manager
   1. Has some programming experience in Python
   2. Worked a little bit in HTML and PHP
   3. Organized coder and planner
   4. Has had experience with debugging code
2. Xiaorong Guo
   1. Has some knowledge of Python
   2. Has experience with coding and code testing
   3. Has experience in interface design
3. Qingwei Kong
   1. Entry level knowledge of Python
   2. Has experience in:
      1. Data analysis
      2. Code testing
      3. Algorithm design
      4. Research
4. Sarah Lee
   1. Some knowledge of Python
   2. Some experience in HTML, CSS, MySQL, PHP, and Photoshop
   3. Worked a little bit in JavaScript
5. Chris Shiohama
   1. Has experience in HTML, CSS and Photoshop
6. Justin Syhongpan
   1. Has practice in HTML, PHP, and BASIC
   2. Is skilled at coding and algorithm design

**Section 2: Management Charts**

**Section 2.1: Project Schedule**

|  |  |  |
| --- | --- | --- |
| **Work Tasks** | **Start Date** | **Duration** |
| **1. Concept scoping** | week #1 | week #1--week #2 |
| 1.1 Meet with customer | week #1 | 0.5 week |
| 1.2 Identify needs and project constrains | week #1 | 0.5 week |
| 1.3 General scope and boundaries | week #2 | 1 week |
| **2. Define function/behavior** | week # 3 | week #3--week #4 |
| 2.1 Define input/output | week #3 | 0.5 week |
| 2.2 Define use cases, functional requirements | week # 3 | 0.5 week |
| 2.3 Research availability of existing software | week #4 | 1 week |
| 2.4 Interface design | week #4 | 1 day |
| 2.5 Estimation of project size | week #4 | 1 day |
| **3.Coding** |  |  |
| 3.1 Part I: TDA Processing | week #5 | week # 5,7,8,9 |
| 3.1.1 Upload TDA | week #5 | 1 week |
| 3.1.2 Convert TDA to text file | week #5 | 3 days |
| 3.1.3 Parse TDA | week #5 | 3 days |
| 3.1.4 Build a link to CSUF portal | week #7 | 3 days |
| 3.1.5 Build a tree of core major courses | week #7 | 1-2 weeks |
| 3.1.6 Choose a track | week #7 | 1.5 -2 weeks |
| 3.1.7 Save/Print info | week #8 | 3 days |
| 3.1.8 Access prerequisites/graduation requirements/class  recommendation | week #9 | 1 week |
| 3.1.9 Help Button | week #9 | 3 days |
| 3.2 Part II: website | week #5 | week # 5,7,8,9 |
| 3.2.1 Set up webserver | week #7 | 2 weeks |
| 3.2.2 Django | week #5 | 2 weeks |
| 3.2.3 HTML, CSS (webpage design) | week #5 | 2 weeks |
| **4. Reports** | week #6 | week # 6,11 |
| 4.1 First report | week #6 | 1 week |
| 4.2 Second report | week #11 | 1 week |
| **5. Testing and debugging** | week # 10 | week # 10, 11, 12, 13 |
| **6. Presentation** | week # 14 | week #14 |
| **Total** | **14 weeks** | |

**Section 2.2: Project Resourcing**

1. Programming Languages: Python, HTML, CSS
2. Software Tools: Django, GraphViz, PDF Miner
3. Systems: Linux Operating System
4. Computing Environment: Digital Ocean Server, GitHub, Laptops

**Section 2.3: Budget**

For budgeting, we will be using the Lines of Code (LOC) method.

LOC (lines of code) Estimation Table

How to calculate pm (person-month):

time to finish the project \* # of team member

So we got 3 months \* 6 people = 18 person-month

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | An average productivity for a system of this type is 37.5 LOC /pm.  Labor rate: $1000 per month  The cost per line of code: $ 27  Based on the info above, the total estimated project cost is $18,225 |
| **Function** | **Estimated LOC** |
| Upload TDA | 30 |
| Convert TDA to text | 5 |
| Parse TDA | 300 |
| A link to CSUF portal | 5 |
| Build a tree of core major courses | 100 |
| Choose a track | 80 |
| Save/Print info | 35 |
| Access prerequisites | 30 |
| Check graduation requirements | 30 |
| Help Button | 30 |
| View a recommanded 4 year class distribution | 30 |
| Total estimated line of code | 675 |

**Section 3: Product Scope and Vision**

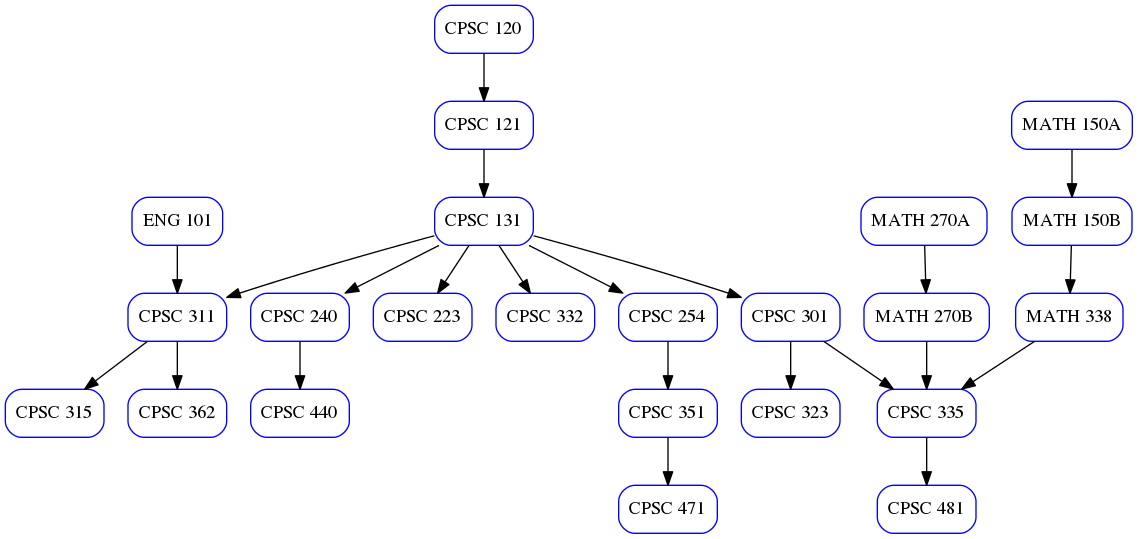
The purpose of this project is to show CSUF students what classes need to be completed and what requirements they need to finish for graduation. The software does this by uploading the student’s Titan Degree Audit (TDA) and translating it into an easy-to-read user interface.

Boundaries of the Project:

This software is a web-based application that needs an internet connection. Currently, it targets Computer Science students at California State University, Fullerton. It shall read in a PDF file of the TDA and translate it to a text file. Then, the program shall extract the information from the text file to display a tree of classes. The tree shall be structured on prerequisites and the classes will be color-coded based on the progress status (Taken, In Progress, Not Taken). The student shall be able to choose their elective track, and it will be extended on the tree. Otherwise, students shall be given an option of “undecided” for the track. The website shall have tabs for displaying the prerequisites, graduation requirements, GE classes, and class distribution. The student will have an option to either save their information to a PDF file, print it, or email it to themselves. The system does not save the student’s data on its server. The information is deleted directly after exiting the program. The user shall be provided a help button that will display an operational manual. The software shall not ask any input from the user other than the TDA. If the system is idle for 30 minutes, then the displayed information will be deleted and the user will be taken back to the home page. The vision for this project is that it will be compatible for all CSUF majors and not limited to Computer Science. We also envision this project to be integrated with the school database so that the student does not have to upload their TDA.

**Section 4: Use Cases**

**Use Case 1: Upload TDA**

**Preconditions: Internet connection  
Description (Steps):**   
1. The user shall click on the browse button.   
2. Then, s/he will choose the PDF file of their TDA.  
3. Click Upload.

**Use Case 2: View Tree**

**Preconditions: Use Case 1  
Description:**  
After the user clicks on Upload, a tree of their core major classes will appear in a tree. Classes will be color coded based on the status (Taken, In Progress, Not Taken). There shall be tabs on the side to see the rest of their TDA information.  
**Alternatives:**  
The user shall be able to choose their track and add it to the tree.

**Use Case 3: Access Prerequisites**

**Preconditions: Use Case 1  
Description:**  
1. The user clicks on the Prerequisite tab.  
2. A list of all major and non-major prerequisites is displayed.

**Use Case 4: Access Unit Requirements**

**Preconditions: Use Case 1  
Description:**  
1. The user clicks on the Unit Requirement tab.  
2. A check-list of all the unit requirements for graduation is displayed.

**Use Case 5: Access General Education Classes**

**Preconditions: Use Case 1  
Description:**   
1. The user clicks on the General Education tab.  
2. A check-list of all the GE sections is displayed.

**Use Case 6: View Class Distribution**

**Preconditions: Use Case 1  
Description:**   
1. The user clicks on the Class Distribution tab.  
2. A recommended four year class distribution is displayed for the students.

**Use Case 7: Save, Print and Email**

**Preconditions: Use Case 1  
Description:**   
The user shall have the option to save their information on a PDF file, print it, email the PDF to themselves, or do all three.

**Use Case 8: Click Help Button**

**Preconditions: Use Case 1  
Description:**   
1. The user clicks on the Help button.  
2. A step-by-step guidance manual shall be provided to the user.

**Section 5: UML Diagram**

The most important use case is the first one because it is a precondition for all the other cases.

Upload TDA

Save, Print and Email

Access Prerequisites

Access Unit Requirements

View Tree

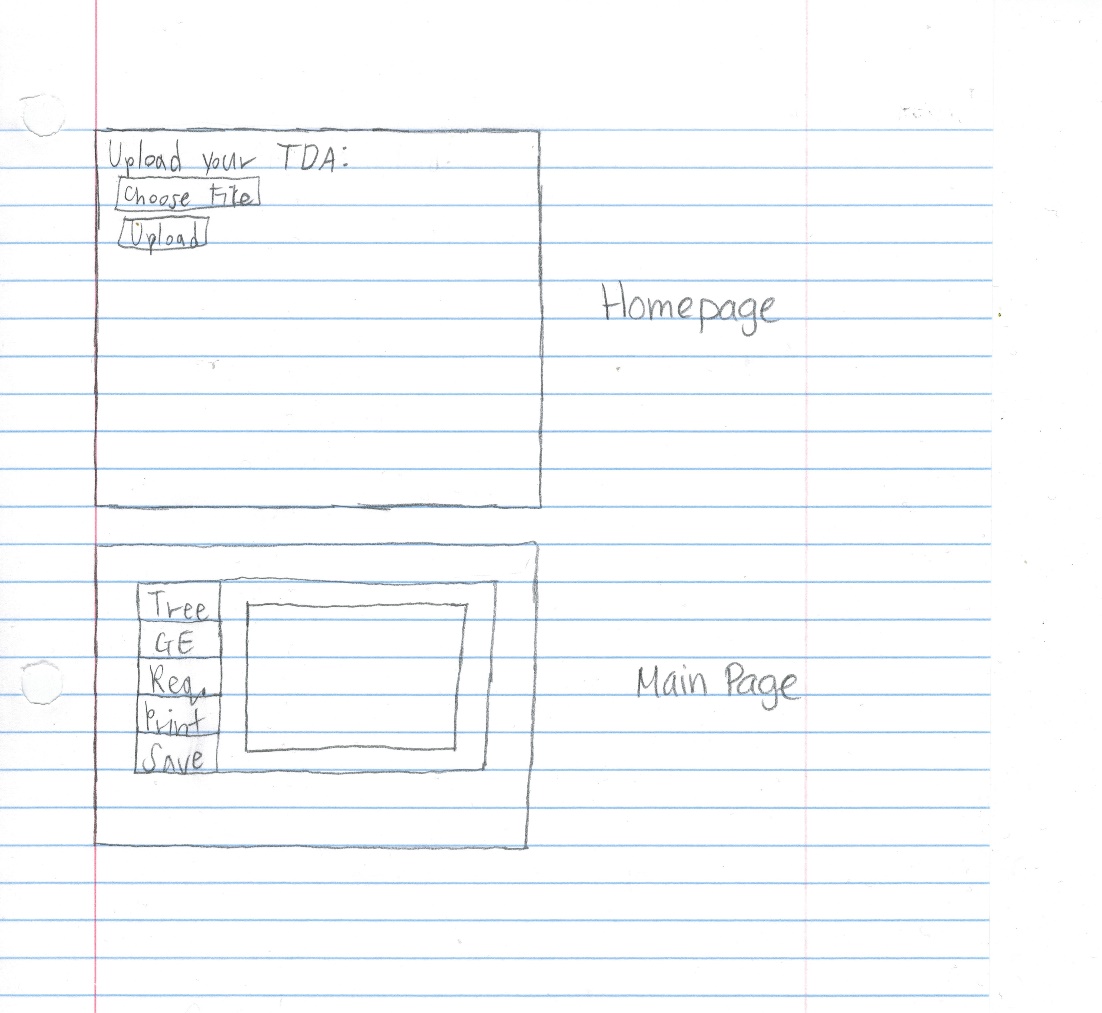
View Class Distribution

Access General Education Classes

Click Help Button

Customer (User)

**Section 6: Interface Sketch**

****

**Section 7: Functional Requirements**

1. The user shall be able to upload their TDA
2. The user shall be able to click on a link to their portal if they do not have their TDA
3. The user shall be able view a tree of their core major courses
4. The user shall be able to choose their track
5. The user shall be able to save their information
6. The user shall be able to print their information
7. The user shall be able to access their prerequisites
8. The user shall be able to check their graduation requirements
9. The user shall be able to view their class status
10. The user shall be able to view a recommended four year class distribution
11. The user shall be able to track their general education classes
12. The user shall be able to email their information to themselves