SW Engineering CSC648/848 Spring 2020

Milestone 01

CsGetDegrees.com

Section 02

Team 04

Submission date: Feb 25, 2020

Team Members:

1. **Team lead: Anh Le - email: ale4@mail.sfsu.edu**
2. **Front end lead: Mohamad Farah**
3. **Front end developer and Github Master: YeeJian(Jeans) Tan**
4. **Back end lead: Aramis Knox**
5. **Back end developer: Timothy Wells**
6. **Back end developer: Russell Azucenas**

I. **Executive Summary**:

Graduation is an arduous process. Students need as much supplementary aid as possible and www.CsGetDegrees.com creates a community apparatus such that each student can contribute to each other's strengths and weaknesses. This is the marketplace apparatus proposed for San Francisco State students. The platform runs on an honor system where students exchange goods and services to benefit each other academically.

CsGetDegrees will help users by offering a platform for them to exchange academic goods and services for money. Sellers will be able to post images and descriptions of their offers. Buyers will be able to browse posted goods and services. Users will have the option to rate one another which will be unique to our service. The rating system helps ensure students of all kinds, the integrity of both our sellers and our buyers.

Our team, for CsGetDegrees, includes bright, geeky, funny and inspired individuals. We all want an A on this project and are therefore more inclined to exceed all expectations. However, we do believe that at the end of the project, we will have developed an application that goes beyond the scope of this course. This goes hand in hand with how passionate, skilled and creative we are when we come together to tackle a common goal.

II. Personas and main Use Cases:

Personas are defined to be the aspect of someone's character that is presented to or perceived by others accordingly. Our first persona for someone who uses CsGetDegrees is a student like Michael. Michael is an average C student who is struggling in class because his attentiveness is low. Michael does not have many available resources to ask for help because he has just transferred to San Francisco State University. The final is approaching, and Michael is becoming increasingly stressed. Michael needs someone with information, experience and resources from someone who has already taken the class.

Michael uses the application to purchase notes from users who have taken a similar course as him. These users have varying ratings therefore varying levels of trustworthiness, and Michael uses these ratings to help him choose the best notes for his money. Other moments Michael searches the application for tutors in his subject and finds one. However, he finds their prices unreasonable, so he messages them to negotiate prices. After finding the users contacts from the app, and within the in app messaging services, Michael finds it convenient to message and deliberate the prices and other details. At the end of using the CsGetDegrees website, Michael walks away with a quality set of notes and a quality worthy tutor. Michael, with the tools he has now, is better prepared to succeed and exceed all expectations placed on him from the beginning of the semester.

Our next persona is Gabriella, a poor but well performing student who is struggling to pay the final portion of her tuition payment plan. Gabriella is an overachiever academically, but her financial circumstances are beyond the scope of financial aid. She's looking for work and hasn't had any luck because she lacks a visa. She's tutored in the past and has been told she's an excellent teacher.

Gabriella posted a listing to become a tutor not only to help other students who are struggling academically, but also strengthen her own knowledge as she values her grades very much. She is able to build up her reputation as a friendly, outstanding tutor among the

San Francisco State students. This eases her financial stress as the scholarships are not enough to cover her expenses. However, she was surrounded by high achievers who did not need her help. With this platform she found potential students so that she can focus on finishing college and getting her degree while expanding her helpful skill sets to others who are in need.

Our third persona is Jacob, a middle aged student with a textbook problem. Jacob is not a tech savvy student. In fact, Jacob can immediately be intimidated by technology. Jacob is looking for an easy and simple way to sell his textbook. He expects that the textbook should be very simple to sell. If the process for listing his textbook is too complicated, he is likely going to give up. He has tried to sell his book on other platforms such as LetItGo but sadly Jacob has given up on those due to its complicated navigation system.

Jacob has a textbook for a class he is done taking and doesn't anticipate needing it again. Luckily for him, CsGetDegrees' user interface was designed by an esteemed team of front end developers who engineered it to be as easy to use as possible, thus keeping users like Jacob at high priority. Because it is so easy to get started with CsGetDegrees, Jacob is able to sign up, list his textbook, and finally get it sold once and for all.

Our fourth persona is Christina, Christina is a mother and a dedicated student. Christina works almost every day and therefore has only been able to take night classes. Having to take night classes, Christina has a difficult time finding a responsive babysitter to watch over her child during class hours. Christina can only take night classes on campus because she works during the day to support her family and pay for classes. Prices for babysitters, especially for the evening, can be egregious and difficult to book.

Christina is currently having problems scheduling babysitters with affordable prices. Luckily, a few students on campus with background in childcare are in need of quick money. This provides an opportunity for Christina to hire capable babysitters at a lower price on a short notice. Having done this, Christina would have time to attend her night classes. CsGetDegrees is a community driven and built around the idea of helping one another

exchange services, and our goal is to provide students with the tools needed to achieve their degrees at an affordable and less stressful way.

Our last persona is Emillio, Emillio commutes from the East Bay and drives to campus every day. Emillio has a surprisingly spacious car that can easily fit six people comfortably. Due to Emillio's commute and classes he attends, Emillio finds himself unable to connect with the college life and make friends. Emillio could also use extra cash for his other necessities such as food and gas he purchases every day.

Emillio uses the Web Application to connect with other daily sfsu commuters around the Bay Area who are in need of a ride. Bay Area prices for rides from other services can be expensive for many students. Bay Area public transportation can be slow as well as long for many students. Emillio solved their problem by offering a ride using our web application. This provides his customers a way to quickly freely and securely commute to school with the help of his service. Emillio uses the Web Application to not only ease his commute but also other students who commute.

III. List of Main Data Items and Entities:

General Users: Users that have not logged in. General users can browse items, but they cannot sell items, buy items, send messages, or leave ratings until they register or log in.

Approved Users: Users that have logged in and have full access to the site and its functions. Needs to login using sfsu email and C'sGetDegree's password. Approved users can post new items to sell, buy listed items, rate other users, and send messages to users through their product listings. Approved users are also given a set amount of imaginary money in order to demonstrate what transactions would look like if we actually hooked up the website to a payment api. However, there will be no way to enter cc info or load any real money into the app.

Admin: Admins are personally-approved users that have all the abilities of an approved user as well as the ability to add departments and courses. Admins also monitor every item that goes live to ensure that they are not inappropriate. Admins can delete listings or users that are deemed inappropriate.

Department: Products are organized on the site on a course-by-course basis. Because of this, every department in the school will be listed on the site and every department will also contain its respective courses. In the event that an item is being sold that is not specifically relevant to any department, a "Misc" department will be available.

Course: Products are organized on the site on a course-by-course basis. Each course can be navigated to via their respective department. Courses contain different items sold by users that are relevant to that course. In the event that an item is being sold that is relevant to a specific department but not a specific course within that department, all departments will have a "Misc" course.

Ratings: All users can have ratings. The only way to rate a user is once a transaction has been completed. Ratings include a star rating as well as a written review that is public to anybody looking at an item being sold by that user.

Messages: Instant messages can be sent to users via their product listings. Because of this, buyers can message specific sellers, but sellers won't be able to message buyers unless they initiate the conversation. This helps avoid fishing, spam, and unwelcome messages that are unrelated to the service. Messages between two users are completely private.

Product: The product here can be referred to as an item or a service posted by an approved user. Every product needs to be reviewed by the administrator before it can be posted on the site. The user's information will be published with the product posted for the purpose of enabling communication between potential buyer and the seller. Every product has to be in a course and a department. The user's rating will also be available to be every user looking at the product.

IV. Initial List of Functional Requirements:

1. System *shall* allow approved users to buy products
2. System *shall* allow approved users to buy services
3. System *shall* allow approved users to sell products
4. System *shall* allow approved users to sell services
5. System *shall* allow all general users, to become approved users via a login
6. System *shall* allow all users, even general users, to browse through the site
7. System *shall* allow all users, even general users, to search through the site using a search bar
8. Approved users *shall* have a balance of **fake** currency that can be used to demonstrate how transactions would take place if a payment api were used.
9. System *shall* allow approved users to send messages to sellers through their listings
10. Users that have received a message *shall* be able to reply
11. Users that have become part of an initiated conversation *shall* be able to view the conversation
12. System *shall* allow users to review sellers after a transaction has been completed
13. Admin *shall* be able to add departments to the database
14. Admin *shall* be able to add courses to the database
15. Admin *shall* be able to approve product listing
16. Admin *shall* be able to remove product listing
17. Admin *shall* be able to remove approved users
18. All users *shall* be able to view other users' profiles via their product listings
19. All approved users *shall* have a profile
20. System *should* put well-reviewed users on the "Dean's List" to indicate that they are trustworthy sellers
21. Products *should* have tags that can be used to enhance the search experience on the site

V. List of Non-Functional Requirements:

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team, but all tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be added
10. No e-mail clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
14. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only"* at the top of the WWW page. (Important so not to confuse this with a real application).

VI. Competitive Analysis:

	Chegg	Amazon	Craigslist	C'sGetDegree
Products	+	+	+	+
Services	-	-	+	+
Private mailing	-	-	-	+
Rate My Professor	-	-	-	+
Student Exclusive	-	-	-	+
Browse	+	+	+	+
Search	+	+	+	+
Seller Rating	+	+	-	+
Recommendation Tab	-	+	-	-
Expert Verification	+	-	-	-

VII. High-Level System Architecture and Technologies Used:

- Hosting Service: Google Cloud Platform
- Hosting OS: Ubuntu 18.04 LTS
- Web Server: Nginx 1.14.0
- Database: MySql 14.14
- Backend Language: Python 2.7.17
- Web Framework: Flask
- Frontend: React
- Version Control: GitHub
- Supported Browsers: The technologies used in our project support all modern browsers
- Analytics: Google Analytics

VIII. **Team and Roles**

- Anh Le (Team Leader)
- YeeJian(Jeans) Tan (Front End, Github Master)
- Mohamad Farah (Front End Lead)
- Russell Azucenas (Back End)
- Aramis Knox (Back End)
- Timothy Wells (Back End)

IX. **Checklist**

- ✓ Team found a time slot to meet outside of the class
- ✓ Github master chosen
- ✓ Team decided and agreed together on using the listed SW tools and deployment server
- ✓ Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
- ✓ Team lead ensured that all team members read the final M1 and agree/understand it before submission
- ✓ Github organized as discussed in class (e.g. master branch, development ranch, folder for milestone documents etc.)