**SW Engineering CSC648/848 Section 01 Fall2016**

**Gator Lodge**

**Team 7**

Lijie Zhou (Team Lead)

Anna Sever(Tech Lead)

Theofanis Koutoulas(Front End)

Ivan Marchenko(Full Stack)

Logan Figgins(Back End)

Hao Xian Zheng(Back end)

**“Milestone 1”**

10/05/16

Version: 01

**1.** **Executive Summary**

GatorLodge is a website that helps SFSU students to rent an apartment in an efficient and user-friendly way. In order to keep GatorLodge a secured community, only authorized SFSU students can rent on the site. However, everybody can register an account and post their apartment. Additionally, we provide efficient filters to make students’ searching easier, eg. students can view apartments by viewing within a specific area, price range, etc. Lastly, GatorLodge is a user-friendly site. We evaluate our sites by steps that first-time users have to accomplished before finding a desirable place. By using the scientific way to evaluate user experience, we make sure that GatorLodge is a user-centered community.

**2.** **Use Cases**

Four primary use cases for Gator Lodge are a browser, landlord, tenant, and administrator. A browser is someone who is looking for an apartment listing site to commit to. A landlord is trying to rent out property to a student. Tenant’s are people who have accounts and are students trying to find a place to live. Lastly, the administrator is making money off of GatorLodge and running the website.

**Browser**

Timothy is an incoming student to SFSU and needs a place to live. He has a reasonable computer skill level so he is comparing different apartment search sites to find his home away from home. His parents are aiding him in his apartment search and he is looking for a website that can cater to all their needs in his apartment search. Timothy is able to browse through listings based on a budget, look up the location of an apartment, and view pictures of an apartment he’s interested in renting. He is able to save apartment units he’s interested in living in so he can discuss them with his parents.

**Landlord**

Ms. Watson has been living in her duplex for thirty years. She is looking to rent out the upper unit and is looking for an easy-to-use website to showcase her unit and rent it out. She uses an easy system that lets her post pictures of her apartment, fill out availability dates, set the location, and manage payments. Ms. Watson does not have a great relationship with technology but she feels obligated to post her unit online in order for people to see it. She is hoping to find something that is easy for her post and manage her listing while searching for a tenant and to let her manage rent easily without leaving her confused.

**Tenant**

Joffrey is a returning San Francisco State University student who is looking for a different place to live. He’s very comfortable with using computers and all sorts of technology, especially user friendly web sites. Since he’s a returning student, Joffrey has an idea of where he wants to live. He is able to search for apartments by using the zip code that school has, which helps with his search. Joffrey being technology sound is able to browse listings and add those listings to his favorites, check reviews, and also look at pictures. He also compares prices so he can find the best price for his apartment, since his parents are paying for his school and board. Seeing how he is a student and school starts at a certain date he is able to check the availability of the listings.

**Administrator**

Anastasia a former San Francisco State University student is able to manage most of the tasks with running the apartments site. Having basic computer skills she is able to log into the site and manage everyday things. She is able to see listings, censor listings, pictures, and also users. She uses the database using the Workbench interface to manage all the data that is provided. Anastasia is making the site a nice and friendly place for students to come and find the apartment right for them. With that she is able to ban accounts, reviews, pictures and listings. She is also managing the money that is being used on the site, since she is making money from the site. Anastasia is able to do all these things in a timely fashion, not spending hours trying to log in and check the site.

**3.** **Data Definition**

**Browsers:**

Only be able to browse postings and images. Cannot post or rent.

**All Users:**

Refers to Member Users as well as Users

**Member:**

Able to post listings/photos and able to rent. Needs to login/register

**Administrator:**

Access to all user data, able to add or delete listing, spam control, able to login

Has full access to database of members and data

**Photos/Uploads:**

Uploaded by registered members and manageable by admin

Unregistered Users shall be able to browse listings, registered users shall be able to login with username and password and the database will keep track of user data such as numbers of listing, last login, and post listings. Admin shall be able to login and browse all user profiles and has the power to ban users and add or remove listings. All registered users has permission to upload photos and post listings. Admin would have final say in whether user uploads has permission to uploads the photos or to post listings

**4. Functional Specifications**

1. All Users shall have the option to browse listings with or without logging in
2. Member Users shall be able to create listings on the websites
3. Member Users shall have the option to upload an arbitrary amount of images to listings they own
4. Member Users may of a listing shall be able to edit their listings at any time
5. Administrators shall be able to ban user accounts who fail to adhere to the terms of service agreements
6. Administrators shall be able to remove images deemed inappropriate
7. Administrators shall be able to remove images deemed inappropriate
8. Users shall have the ability to authenticate their identity using facebook login
9. Users shall have the option of registering without facebook becoming a Member User
10. All Users shall have the ability to search through listings with various filters
11. All Users shall have the option to sort listings on the screen based on various conditions
12. All Users shall be able to view the website on various size screens easily due to the site’s use of media queries to determine the screen size being used
13. Member Users shall have option to add potential listings to their wish list so that they can be easily viewed at a later date
14. All Users shall be able to view the location of the listing using embedded google maps services
15. All Users shall be able to visually determine the bus routes to and from San Francisco State University
16. Member Users shall be able to review listings they have previously lived in with 1 to 5 star rating
17. Member Users shall be able to leave written reviews that can be viewed by other users in addition to a star rating if desired

**5. Non-Functional Specifications**

* System shall respond visually within 5 seconds
* File size in no time shall exceed 2 Mbytes
* Users with high-school diploma, after 1 hour training, shall complete the task in 5 minutes with no more than 2 errors.
* Each WWW page shall have official company logo in upper left corner
* The following user data shall be collected and stored; the data shall be used ONLY for record keeping
* Application shall be developed using class provided LAMP stack
* Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks shall be explicitly approved by Marc Sosnick on a case by case basis.
* Application shall be hosted and deployed on Amazon Web Services as specified in the class
* Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. It shall degrade nicely for different sized windows using class approved programming technology and frameworks so it can be adequately rendered on mobile devices
* Data shall be stored in the MySQL database on the class server in the team's account
* Application shall be served from the team's account
* No more than 50 concurrent users shall be accessing the application at any time
* Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
* The language used shall be English.
* Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
* Google analytics shall be added for major site functions.
* Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.
* Pay functionality (how to pay for goods and services) shall be simulated with proper UI, no backend.
* Site security: basic best practices shall be applied (as covered in the class)
* Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development, and only the tools and practices approved by instructors
* The website shall prominently display the following text on all pages "SFSU/FAU/Fulda Software Engineering Project, Fall 2016. For Demonstration Only". (Important so as to not confuse this with a real application).

**6.** **Competitive analysis**

*(+) means the site does positive on the feature and (-) means the site does negative.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Target | Content | Search/efficiency | User Experience | | Security |
| GatorLodge | SFSU | (+) | (+) | (+) | | (+) |
| Craigslist | General public | Busy interface, all words. Looks like a documentation.  **(-)** | The filters are really confusing: search titles only, has image. What do these even mean? Who do not want to see image before renting?  **(-)** | A lot of words and information is not well-organized and sometimes misleading.  **(-)** | | Support HTTPS  **(+)** |
| Rent | General public | 1. Very simple but clear interface.  2.There is a moving quote function that helps students with moving service.    **(+)** | 1.Four filters: location, min price, max price, beds.  2. More filters with advanced searching  **(+)** | Registration is very fast, support both google and FB log in.  3. Icons are clear (eg.wishlist) and college students friendly (eg. site logo).  **(+)** | | The site doesn’t support HTTPS  **(-)** |
| Places  For  Students | College students | Unrelated information at the spotlight of the site.  (-) | The site offers three ways to search school but it doesn’t make a lot of sense: search by letter, enter name, map search.  (-) | Very bad user experience: login is in the middle of the main page and on top of a picture. Also, same information has many different entries for access.  (-) | | Support HTTPs  (**+**) |
| College  Student  Apartments | College students | 1.Very easy to start the search and add listing.  2. It is redundant to add all partner schools on the main site since the search already covers school searching.  (-) | Search bar supports school name and postal code. It also has an auto-fill/correction feature.  (+) | The site is very visually pleasant. The registration is easy, but doesn’t support Google/FB log in. Listers have two options (either list a whole apartment or sublease). (+) | | This site doesn’t support HTTPS  (-) |
| For  Rent  University | College students | This site gives guidance to three groups of people: students, parents, and landlords. Each group get advice before they start their search. Landlord are also able to pick students based on their preference, eg: BA, MA, International.  **(+)** | Simple search bar based on college name and postal code. It also has auto-correction feature. Later, more advanced filters are added, eg: price range, bedrooms, bathrooms, etc.  **(+)** | The site flow is very intuitive. On each apartment page, the information is well-organized and the layout is very consistent.  **(+)** | | Support HTTPS  **(+)** |

**7. High-level system architecture**

7.1 Overview.

This will be a monolithic software application served over www built using popular open source tools. We will use LAMP stack on the backend with MINI PHP framework as a backbone MVC design paradigm. Front-end will be traditional mix of HTML, CSS, JavaScript. To address cross-browser compatibility issues, we will utilize “normalization” css and javascript libraries.

7.2 Top Considerations.

* Make things as simple as possible and avoid unwarranted complexity.
* Follow MVC pattern to decouple business logic from data and presentation layers.
* Utilize OOP principles to encapsulate features and type less code (code reuse).
* Modular loosely coupled components.
* Intra-system cohesion.

7.3 Tools.

The required open source software we shall use: MINI php framework, jQuery, jQuery UI, any jQuery plugins that make our life easier, Twitter Bootstrap, and any Google APIs. All other external software libraries require explicit approval in writing from Mr. Marc.

On the organizational side of things, we must use GitLabs as our shared code repository system. Use our unix accounts on Amazon S3 server as sandbox to deploy and test our code changes prior to committing CLs to GitLab. Lastly, we are expected to use MySQL Workbench as the main DB management tool. All DB related work will be done directly via Workbench client.

7.4 Model Layer.

MySQL will be used as primary data storage and data management solution. We shall design optimal schema that makes sense within project context. Probably, we will also use DB to store binary content such as images, since this will be easier than introducing file system management approach. Also, we should utilize single DB bridge/connector that will be used by application for all DB related I/O.

**8. Team**

Team Lead: Lijie Zhou

Tech Lead: Anna Sever

Team members:

Frontend: Theofanis Koutoulas, Anna Sever

Backend: Logan Figgins, Hao Xian Zheng, Lijie Zhou

Full-stack: Ivan Marchenko