

Milestone 3 More Detailed Requirements, Architecture, and a Vertical Software Prototype

OWL Alerts

Panda Development Team (Group 7)

Project Members

Product Owner: Paul Birns

pbirns2016@fau.edu

Scrum Master: Danica Brozowski

dbrozows@fau.edu

Back End Lead: Akiva Green

green2015@fau.edu

Front End Lead: Aaron Zygala

azygala2016@fau.edu

Advisor

Shihong Huang

shihong@fau.edu

Documentation Date: 4 November 2019

Florida Atlantic University

CEN 4010 Principles of Software Engineering, Fall 2019

History Table

Revision	Revision Date
Paul wrote Data Definitions	9/18/2019
Danica wrote functional/Non Functional Requirements	9/17/2019
Aaron wrote Executive Summary, Competitive Analysis	9/20/2019
Akiva wrote Overview of Use Cases, and High-level System Architecture	9/20/2019 - 9/21/2019
Danica expanded and updated functional and nonfunctional requirements	10/25/2019
Akiva updated High-level System Architecture with Database Organization	11/1/2019
Aaron constructed UML Diagrams	11/3/2019
Akiva finished vertical prototype	11/3/2019
Danica Updated data definitions	11/4/2019
Paul updated the Data Definitions and wrote key risks	11/4/2019

NOTE: For this milestone, our team decided to work with Google Docs. Instead of utilizing GitHub, our team decided it was best to work through Google Docs to avoid constant merge conflicts or delayed editing. Google Docs is much faster, and allows for simultaneous edits. While working together through a phone call this document allows for cohesive teamwork.

You can view/comment on this document with the following link:

<https://docs.google.com/document/d/1GyRd5phKnfrIHUmrIHp01Z09bd0euNKK9OM48NpXbe0/edit?usp=sharing>

Link to Main Page:

http://lamp.cse.fau.edu/~cen4010fal19_g07/

Link to Vertical Prototype:

<http://lamp.cse.fau.edu:14125/>

I. Executive Summary

The Florida Atlantic University Alerts System (shortened to OWL Alerts) will give students the ability to alert the campus staff whenever an issue has occurred as well as alert students to events taking place around campus. On campus, students can let the campus know that there is a water leakage at certain locations of the campus, so that the staff can quickly fix the issue. Additionally, commuting students can receive alerts about traffic and campus parking before travelling to their classes. The Alerts system can help make a student's life simpler, so they do not have to worry about minor issues and can focus more on their classes and studies. To use the system, students will need to create an account, at which time they will be able to view a dashboard about upcoming campus events and issues. They can choose to receive daily or weekly emails to keep them updated about school-wide events, or any issues occurring on

campus. When students create an account, they will specify whether they live on campus or not. Students that live on campus will receive notices and messages about their specific buildings, and off-campus students will receive traffic alerts and parking notifications. Students also have the ability to report specific issue posts if they are inappropriate or untrue. The system is simple and easy to use, with buttons and functions easily laid out on a user's cell-phone or computer.

We have discussed the user's view of the software. Administrators, such as faculty in charge of cleaning the campus, or members of the IT department, will be able to create accounts as well. An administrator's account will be able to view the current list of issues, including the date the issue was created and resolved, the description of the issue, and the creator of the issue. They can also see the specific administrator who is designated to fix the issue. Administrator's also have the ability to remove any inappropriate/untrue issue reports from the backlog. The administrator view of the application is useful for faculty members who need guidance about the location and description of specific issues.

II. Competitive Analysis

The competitors focused on in this competitive analysis are other systems to allow students and campus faculty to communicate. This includes the current alert system used by FAU and systems used by other colleges.

OWL Alerts' Features	Competitors' Features(FAU current system)
Can receive campus alerts, parking notifications, and school event reminders all in the same	Spreads these alerts across multiple different systems.

place.	
Allows students and campus faculty to communicate through an interactive platform.	Allows students and campus faculty to communicate via email notifications.
Allows students to receive email notifications for alerts.	Allows students to receive email notifications for alerts.
Gives students a nice user experience, with a dashboard outlining upcoming events, current issues, and parking information.	Sends email alerts and phone calls for different issues. No real user experience.

The above table displays the competitive relationship between our proposed software and the current systems used by FAU and other schools. Clearly, our system is focused on a great user experience and system that allows for clear communication between the faculty and students at FAU. This can allow students to feel more welcome and calm during a stressful school year. This system also gives students an immediate answer to their issues, as administrators will be issued to aid their problems almost immediately. Additionally, this system is constantly updated with parking information, giving off-campus students help as well. This system aims to serve as an all-encompassing FAU information and campus issue hub, to help and aid students in any way it can.

III. Data Definition

- User - A person who uses the website

- User Types - the different types of users who will use the website
- Student User - A type of user who is a student
 - ZNumber - The student user's school identification number
 - Class Status - Identifies the class level of the student (freshman, sophomore, junior or senior)
- Commuter Student - A student who takes classes at the university, but does not live in a dorm
- Dormer Student - A student who takes classes at the university and lives in a dorm
- Admin - A person who uses the website and is responsible for maintaining issues that appear on the website
 - Has post removal capabilities if a post is reported inappropriate or inaccurate
- Garage - One of 3 car garages on the FAU campus that students and faculty use to park their cars.
 - ID - identifies garage by a unique number
 - Name - identifies garage by a unique name
 - Location - Street name on FAU campus where a particular garage is located
 - Full - indicates if a particular garage is full or not
 - Total spots - The number of total spots that a garage has.
 - Free Spots - The number of available spots available for use if any
- Issue Backlog - A list of problems on campus that an Admin is responsible for handling (e.g. flooding on campus)
 - All Issues - All the issues that were reported

- Pending Issues - All issues that need to be resolved
 - Resolved Issues - All issues that were resolved
- Faculty - A person who works for FAU
- Issue List - A list of all the issues that were created
- Event List - A list of events happening on campus
 - Event Location - Location on campus where event is taking place
 - Event Description - Description indicating what the event is about and what will be taking place at the event.
 - Event Start Date - When an event will begin
 - time and date
 - Event End Date - When an event will end
- Time and Date
- Issue Status - This issue can be marked as resolved or opened
- Issue Location - The location on campus on where an issue has occurred
- Issue Verified - The status on if this issue was verified
- Issue Resolved - The status on if this issue was resolved
- Residency Status - Identifies the type of residence that user holds
- Building Name - Name of a building on campus (e.g. General South)
- Room Number - Identifies the room number of a particular building
- Report an Issue - The issue that can be reported
 - Issue Title - The title of the issue
 - Issue Description - The description of the issue

- Add Report - An optional feature where Admins can add a formal report on an issue or event.
 - Only pdf accepted file type
- Issue Panel - A panel that shows the list of issue
- Admin Assigned - Admin assigned to monitor issue
- Profile Pane - The panel where the user can login or logout of the website
- Image - An image that will be displayed on the website. The expected file format will be a JPEG file with a max size of 20 MB.
- Sign In - A page on the website that will be used to authenticate a user
 - Email - the identification used to identify a unique user
 - Password - the user's created password to authenticate themselves
- Sign Up - A page where the user can register themselves on the website
 - Name - The user's full name
 - Email - Email issued by FAU to students and faculty
 - Password - The user's created password
 - Phone Number - The user's phone number
 - Address - An address of a location/mailing in the US
- Sign Out - A method for the user to sign out of the website
- View Profile - The ability to view current information the user posted on the website
 - Your Profile Information - The main area on the website where the user can choose to update their information
 - Name

- Email
 - Phone Number
 - Address
 - Dorm - The user can change the dorm that they currently reside in
 - Update Information - A button that the user can press to update all their profile information
- FAU Secure Login - The area on the website that the user will be

IV. Overview, scenarios, and use cases

The Florida Atlantic University Alerts System, OWL Alerts, is developed for use by the average FAU student and faculty.

Use Case 0: Log-In:

1. The user accesses the website.
2. The system displays the message:
 - a. Please enter your username and password to continue.
3. The user enters their account information
4. The system displays the menu:
 - a. Choose “Post an Issue” to communicate an issue with the campus.
 - b. Choose “View your Dashboard” to see current events, issues, and notifications.

- c. Choose “Log-Out” to leave your account.

Variation #1: Admin Log-In:

1.1 In Step 3, the admin enters their account information.

1.2 The system displays the menu:

1.2a Choose “View Verified Issues” to look at all faculty verified issues

1.2b Choose “View Unverified Issues” to look at posted issues from students that have yet to be verified by faculty.

1.2c Choose “View Resolved Issues” to look at issues that have been resolved.

1.2d Choose “Log-out” to leave your account.

Initial Assumption: Existing user wants to login to check if there are any issues or events on campus they should know about. Then log out.

Normal : User logs on and logs out with no errors

What could go wrong: User forgets his username or password. The user can follow a link to retrieve his username via email. The user can follow a link to reset his password via email.

Other activities: events and issues are being updated and reported by users.

System State on Completion: User logs out

Use Case 1: Post an Issue

1. The user carries out Use Case: Log-In.
2. The user chooses “Post an Issue.”
3. The system displays an empty textbox, with an option to “Post the Issue.”
4. The user enters their issue, and any images that correspond with the problem.
5. The user posts their issue to the system, and they receive a confirmation message saying “Thank you! Your post has been received.”
6. The system saves the post to the Issue Backlog.

Initial Assumption: There is an issue on campus the user would like to report and inform the students and faculty about.

Normal: User selects “post an Issue” and uploads photos and details related to the issue.

What could go wrong: Issue is already reported by another user. Second post is reported by another user as a duplicate and is taken down.

Other activities: Admin of the platform takes down second post.

System State on Completion: Pos is completed and duplicates are eliminated

Use Case 2: View the Dashboard

1. The user carries out Use Case: Log-In.
2. The user chooses “View Your Dashboard.”
3. The system displays a menu:
 - a. Press “View school wide events” to see events occurring on campus.
 - b. Press “View your dorm’s issues” to view recent problems in your dorm building.
 - c. Press “View major issues” to see the major problems across campus.
4. The user chooses “View your dorm’s issues.”
5. A neat list is displayed on the screen, showing the recent issue posts at the user’s dorm building.

Variation #1: Commuter Student:

1. Start from Step 3.
2. The system displays a menu:
 - a. Press “View school wide events” to see events occurring on campus.
 - b. Press “View parking information” to find open parking spaces on campus.
 - c. Press “View major issues” to see the major problems across campus.
3. Since the user commutes to school, they can view parking information instead of dorm issues.

Initial Assumption: The user wants to login into the website to view information pertaining to their status as a commuter.

Normal: The user is able to login into the website see information related pertaining to their status (e.g. school events, parking information major issues).

What can go wrong: The user account does not exist. The user does not have the appropriate permissions to view their issues. The user account is not confirmed by the school. The user no longer attends the university, so an account is no longer activated.

Use Case 3 - Faculty Verifying Issue:

1. The faculty member carries out Use Case: Log-In (Variation 1: Admin Login)
2. The faculty member chooses the “View Unverified Issues” option
3. The faculty member clicks on the issue that has been verified by themselves and hits the verify button.
4. The faculty member can click the edit button to edit the post for when it is viewed on the “Verified Issues” page.

Initial Assumption: The user wants to login into the website to view “Unverified Issues”.

Normal: The user is able to login into the website and view the “Unverified Issues”.

What can go wrong: The user account does not exist. The user does not have the appropriate permissions to view the unverified issues. The user account is not confirmed by the school.

Use Case 4 - Faculty Resolve an Issue:

1. The faculty member carries out Use Case: Log-In (Variation 1: Admin Login)
2. The faculty member chooses the “View Verified Issues” option
3. The faculty member clicks on the issue that has been resolved.
4. The faculty member clicks the “Resolved” button to mark the issue as resolved.
 - a. The issue can now be viewed on the “Resolved Issues” page.
 - b. The faculty member can choose to notify individuals that are affected (users in the affected dorm, affects a large body of users, and etc.)

An FAU faculty member who is a part of the campus IT team has been briefed on a new web application that his department will be using to track student reported technology issues. He creates his account via the verified faculty sign-up page. After account creation he logs in and sees the list of issues (unverified), verified issues, and resolved issues. Starting with unverified issues they will look through them and based upon their judgement they will verify the issue physically. Once they have verified the issue they will mark the issue as verified and delete all the duplicates. Upon verification the faculty member will move to have the issue resolved. Once the issue has been resolved it will be marked as resolved.

Initial Assumption: An issue has been uploaded by a user, or a group of users. An admin has logged into the website, and is allowed to view the issue backlog. A specific issue has already been resolved by a member of the faculty.

Normal: The admin chooses to view the verified issue backlog, and chooses the specific issue that has been resolved. The admin chooses to mark the issue as “Resolved,” which is then added to another list of “Resolved issues.” Additionally, the admin has the ability to notify affected students, such as members of the specific dorm building, etc.

What can go wrong: An issue is incorrectly marked as resolved, if the admin makes a mistake. Once an email is resolved, other admins are emailed to ask if it is marked correctly. If it is not actually resolved, the other admins can prevent it from being marked as resolved.

Other activities: Other admins can be logged into the site, and approve resolved issues as they are marked.

System state on Completion: A confirmation message letting the admin know that the issue has been sent to other administrators to confirm it is resolved, the issue’s current status is “awaiting approval.”

V. High-level functional requirements

A. Usable from major browsers such as Chrome, Firefox, and Safari

1. Level 1 priority
2. Formatting of the application should be responsive for the three listed major browsers on desktop and mobile devices.
3. Users must be able to easily navigate application pages and view information clearly and readily.

B. Read and write information about events and problems with text and photos

1. Level 1 Priority
2. Example: 9/16/19 water tank explosion 3rd floor IRT
3. Software must be able to access files created by external tools and programs.
4. Can accept, store and display files of .jpg .pdf and .png
5. User must identify the type of report being posted
 - a) Ex: Issue, Event , Miscellaneous

C. Edit status of existing posts

1. Level 1 Priority
2. Example : 9/16/19 water tank explosion 3rd floor IRT(resolved 9/17/19)
3. Application updates and status of an event to resolved and moves it to categorized archive.

D. Verify events and problems and status

1. Level 3 Priority
2. Example : 9/16/19 water tank explosion 3rd floor IRT(resolved 9/17/19)
*verified by (# of users)
3. Applications has a verify button users can click and software displays a counter of verified clicks by independent users
 - a) One click per user

E. Ability to add a report to the incident

1. Level 2 Priority

2. Software must be able to access files created by external tools and programs.
3. Documents accepted are PDF only
4. Submitted only by approved party. NOT available for students

F. Archive of past reports by section/type

1. Level 1 priority
2. Seperate database for each type of report
3. Events archive
4. Maintenance Archive
5. Miscellaneous Archive

G. Shareability on user social media

1. Level 2 priority
2. Software should be able to share content to an external social media platform
3. Share buttons for Facebook, Instagram and Twitter will be available to every post

H. Users CANNOT direct message each other

1. Communication between users is strictly limited to comments on user posts.

I. Users will not have access to other user data

1. No profile page specified for individual users.

VI. List of non-functional requirements

1. Product Requirements:

- a. Available 24/7 with database updates in real time.
 - i. Level 1 Priority
- b. Required photo with every post for validation of event/issue.
 - i. Level 1 Priority
 - ii. .jpeg, .pdf and .png accepted only 20MB max size.
- c. Backlog up to a years worth of data.
 - i. Level 2 Priority
 - ii. Viewable on archive page
 - iii. Searchable in search bar
- d. Post reporting capability to report unsafe and inaccurate content to be removed.
 - i. Level 2 Priority
 - ii. Available for all users to report bad content
 - iii. Content reported is reviewed and removed by admins
- e. Informative error messages.
 - i. Level 2 Priority
 - ii. Unreadable file or wrong file type uploads of photos and reports
 - 1. States acceptable file types

iii. Incorrect password/login

1. Displays option for password reset

f. Fast refresh time.

i. Level 1 Priority

ii. Page refresh every 3 minutes

g. Event/problem searchability

i. Level 2 priority

1. Search bar on each page returns matches to searched
criteria

a. Searches entire database of all events/problems

h. Run ads for businesses on camps

i. Level 3 priority

1. Examples chick fila, starbucks etc.

2. Cover expenses for maintenance and housekeeping

2. Eternal Requirements:

a. Secure login/registration interface

i. Level 1 priority

ii. Must be matched and verified to FAU email

iii. Error message when login isn't correct

b. Required agreement to privacy policy and user conduct agreement.

i. Level 1 Priority

ii. Must select agree to terms to register an account to post

- c. Privacy agreement at registration and link available at all times on app.
 - i. Level 1 Priority
 - ii. Must select agree to terms to register an account to post
- d. User agreement and code of conduct available upon registration and linked on site to be accessible at all times
 - Level 2 Priority
- e. Unauthorized use is not allowed
 - i. Level 1 priority
 - ii. No users outside of FAU able to post or edit posts

3. Organizational Requirements:

- a. Secure data collection
 - i. Level 1 priority
 - ii. Separate databases for report classes
- b. Simple user interface developed for users
 - i. Level 2 priority
 - ii. User can learn the in's and out's of the app in under 30 minutes
- c. Delivered by Final project due date
 - i. Level 1 priority

VII. High-level system architecture and database organization

Language(s) Used: HTML, CSS, Javascript, Typescript

Runtime Environment: NodeJS

Framework Used: ExpressJS, jQuery

Tool(s) Used: Visual Studio Code, Brackets, Apache

Supported Browsers: Chrome, FireFox, and Safari

API Used: Google Maps, Google Oauth

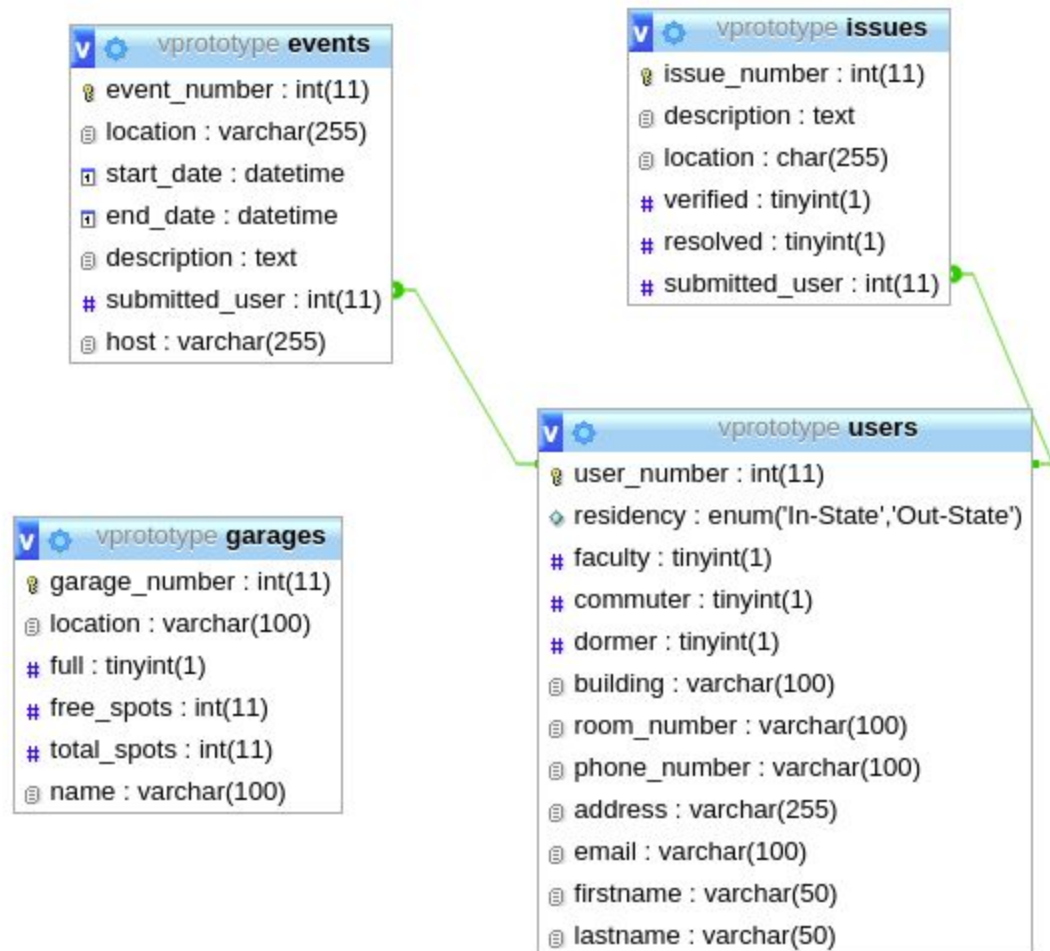
Database: MySQL

Media Storage: any images will be stored in database

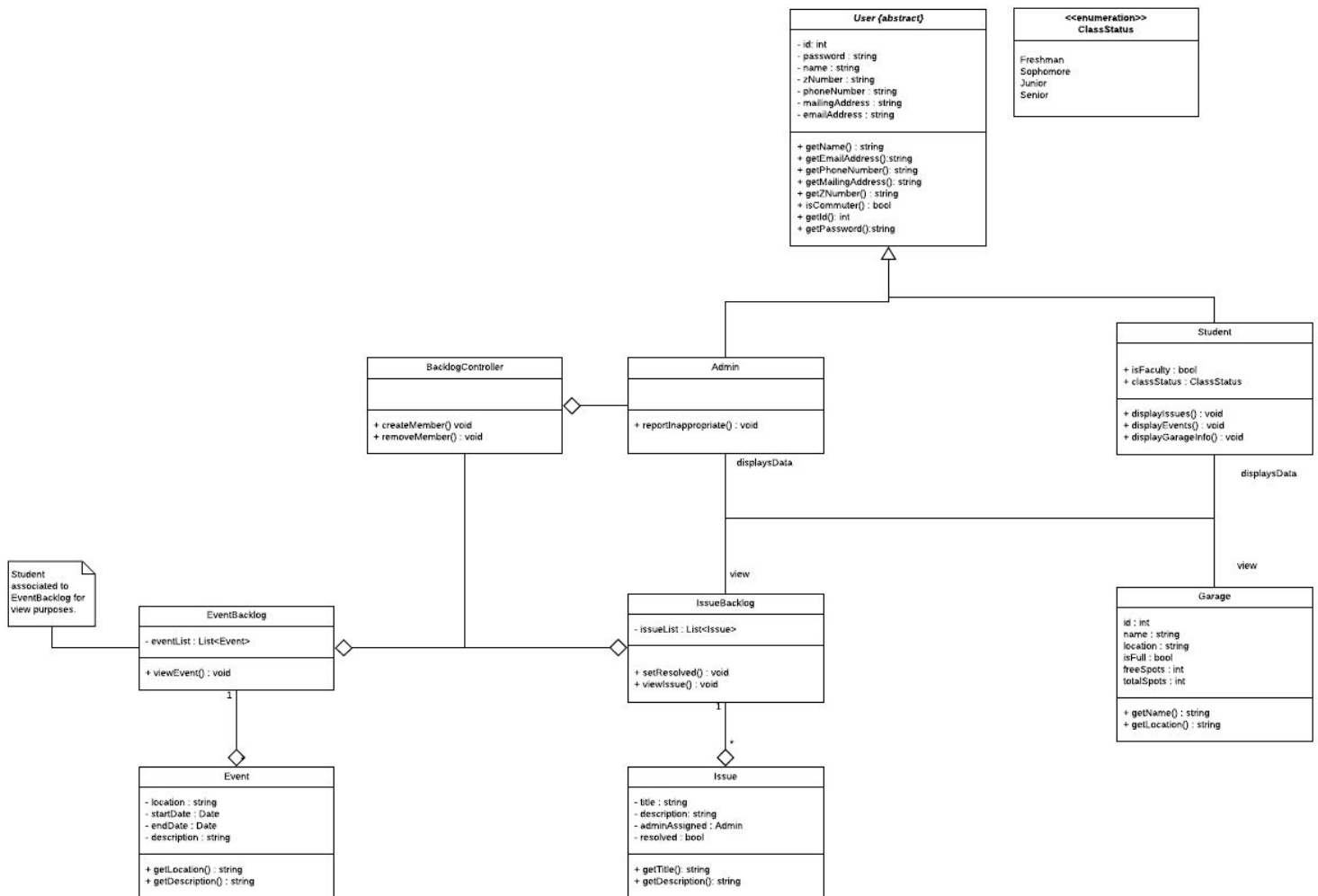
Search/filter architecture: Binary search algorithm , AJAX

The database is organized into 4 tables, two of them have a foreign key that leads to another. Based upon what our website's functional requirements there are a users, events, issues, and garages table for this database. The users table is for all registered users of the application. The events table will be for all events posted to the website. The issues table will be where all issues created will be held. Since a user has to create an issue record it will save the user who created the issue. The garages table will be used for garages on campus holding values for their total number of spots, the number of free spots left and a boolean for if the garage is full.

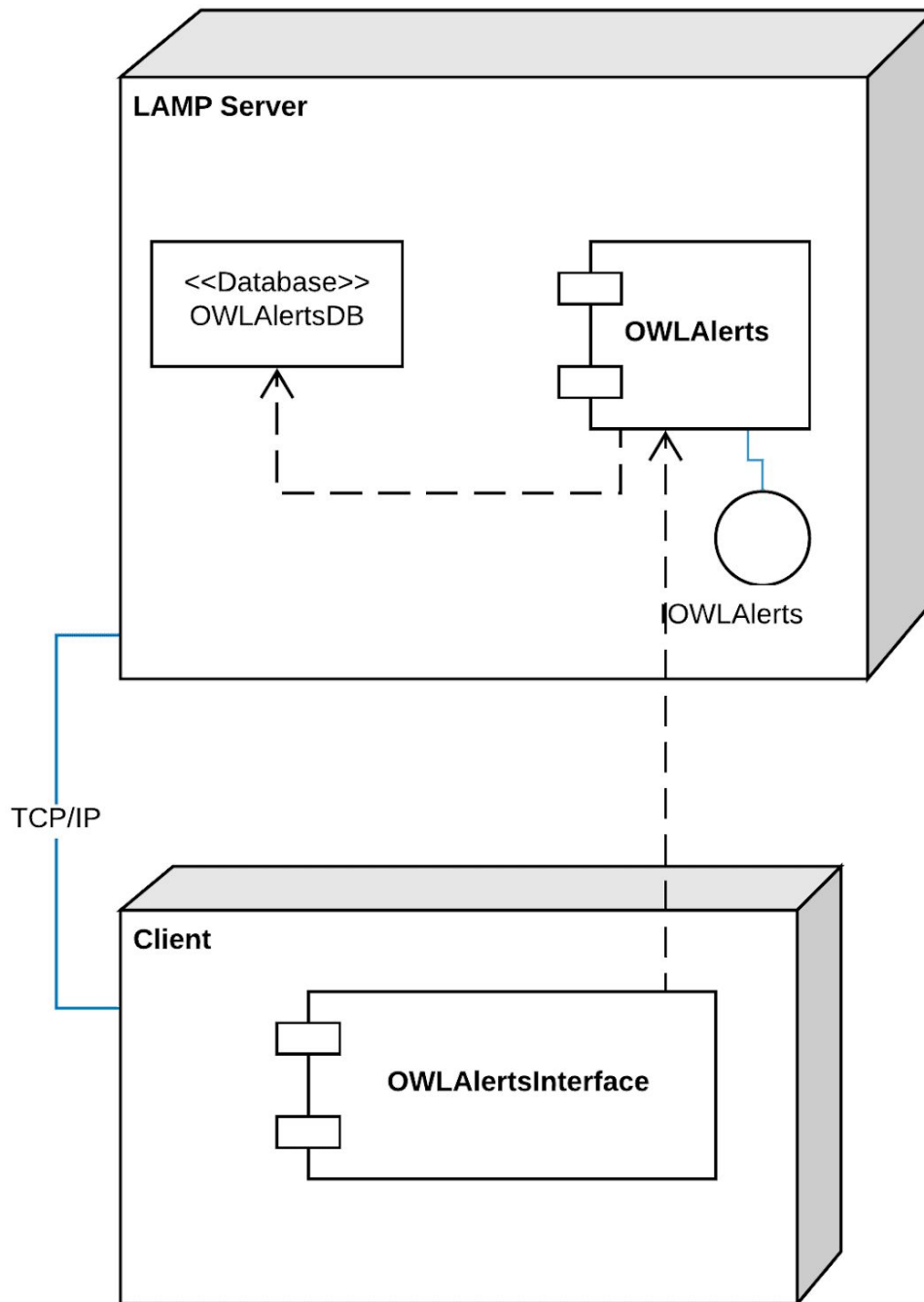
Database Organization



VIII. High-Level UML Diagrams



Component and Deployment Diagrams



IX. Key Risks

1. Skill Risks

a. Risk 1

i. The team members do not have experience programming in NodeJS. NodeJS is a programming language geared towards running on the server. The risk could be defined as the team members not having experience with setting up a NodeJS project.

ii. This risk will be resolved by having all the team members do a tutorial on NodeJS online. There are a numerous amount of resources available online to learn this developmental tool, such as youtube and W3Schools. Each team member will have an opportunity to practice using NodeJS before development starts on the product. The following link will be used to learn about NodeJS:

<https://www.w3schools.com/nodejs/>.

b. Risk 2

- i. The team members do not have experience using the Google Ouath and Google Maps API. These two APIs will be used to pinpoint specific locations on a map on the website. The team members do not have experience in setting up this library on a website.
- ii. This risk will be resolved by having all the team members learn about the Google Ouath and Google Maps API from online resources. The following link is a tutorial on how to set up the Google Ouath API on the web client:
<https://developers.google.com/identity/protocols/OAuth2UserAgent>. The following link is a tutorial on how to use the Google Maps API for javascript based applications:
<https://developers.google.com/maps/documentation/javascript/tutorial>. The team members will learn how to use this API before development starts on the main project.

2. Schedule Risks

a. Risk 1

- i. The ability to complete all the required functionality in the documents by the delivery date. There are numerous items on the list that have to be implemented, and each component takes time to develop.
- ii. We will prioritize our requirements accordingly and sparingly to meet our deadline. We will complete the highest priority first.
- iii. To help resolve this risk further, the Trello board will be utilized to monitor completion of all tasks. Each team member will be assigned a specific task to complete in the Trello board, with each task having a proposed date for completion. If the team member does not meet the proposed date deadline, then that team member will be required to give a formal apology speech to the whole team, explaining why he could not meet the deadline.

3. Technical Risks

a. Risk 1

- i. There could be an issue where the team members are unable to access the required resources to complete the

project. For example, having access to a LAMP server where the project will be hosted.

- ii. This risk will be resolved by having all the team members ensuring that all the resources that will be needed to complete this project is available before development starts. The following is a list of potential problems that may be encountered:

- 1. A resource that is vital to this project is having access to the LAMP Server. This utility is vital as this is what will be hosting the team's project.

- a. If this problem occurs, the team will contact the company responsible for managing the LAMP Server (help@eng.fau.edu), and will work with them to resolve this issue.

- 2. A potential risk is the inability to use the MySQL Database with the NodeJS Framework. The MySQL database is imperative to this project as this database will be responsible for retaining

important information, such as all the user's information.

- a. If this problem occurs, the team will contact the company responsible for managing the LAMP Server (help@eng.fau.edu), and will work with them to resolve this issue.

3. A potential risk is that the LAMP Server will not support an NodeJS application. Since the project will be developed using NodeJS, it would delay development as this project was geared towards using the NodeJS framework.

- a. If this problem occurs, the team will first contact the company responsible for managing the LAMP Server (help@eng.fau.edu), and will work with them to resolve this issue. If they are unable to resolve this issue, we may have to consider using another framework. A

backup framework to use in case it fails is to use Vanilla Javascript.

4. A potential risk is that the NodeJS application is unable to use both the Google Ouath and Google Maps API.

a. If this problem occurs, we will try to figure out a way to be able to use these services. If we are unable to, we will then contact help@eng.fau.edu. If they are unable to resolve it, we may have to resort to using a static google maps image to display the required result. The team could also implement a method to show popup labels on the map, if required.

4. Teamwork Risks

a. Risk 1

i. A team member fails to communicate if he or she is having issues to complete a task. As communication is vital to this project's success, any failure in

communication would mean a delay in completing the project on time.

- ii. To resolve this issue, the team will conduct a daily fifteen minute meeting during development. During that meeting, each team member will be required to answer the following:

1. What tasks that team member was working on?
2. What tasks that team member is planning on doing?
3. What issue(s) that team member is experiencing?

5. Legal Risks

a. Risk 1

- i. A legal risk to consider is the website requiring external images from outside sources. An example of an image that could be required could be a picture of an owl to be on the website's homepage.
- ii. All of the images that will be used on the website will have either Creative Cloud licensing or in the public domain. The main source to retrieve images that can be

used for free without any attribution is

<https://pixabay.com>. In the scenario where an image has to be used with attribution, the image would be posted on the appropriate page on the website, and a reference would be added to that image.

b. Risk 2

- i. A legal risk to consider is the website requiring reference material from other sources. An example could be a quote about a known alert system in the about page on the website.
- ii. Any reference information that is required on the website, will be cited accordingly.

c. Risk 3

- i. A legal risk to consider is the website requiring external software. For instance, a software that the website will be using is the Google Maps API.
- ii. Any external software that will be used will be referenced on the website itself. However, before using the software, the software will be checked for its license,

to ensure that it will be appropriate usage to use on the website.

d. Risk 4

- i. A legal risk that needs to be handled is the website's privacy and terms of use page. These two items will inform the user about what kind of data is collected from them and the agreement that the user agreed upon from using the website.
- ii. The website will include all the required legality information, such as terms of use, privacy policy, etc.

Peer Evaluation

Paul -25

Akiva-25

Danica-25

Aaron-25