Owl Snapshots

Milestone_1 G6 Project Proposal and High-Level Description
CEN4010-F19: Principles of Software Engineering
September 23, 2019

Grant Kveton
gkveton2017@fau.edu

Kevin Fash
kfash2017@fau.edu

Nicole Appleton Guerrero
nappletongue2015@fau.edu

Revision History	Date

Executive Summary

Owl Snapshots is an issue-tracking system accessible to FAU students from their web browser. It makes use of a tiled interface that provides a simplified viewing experience of submitted events. Users will be able to see other issues reported by others so that they can confirm its existence or provide additional details. This is similar to how Waze and Reddit function. Besides being a centralized solution for issue reporting and management, it also acts as a hub for campus events and an informal advice forum used to inquire about general campus-related information. General users will be able to create a snapshot of an event. Other users can then see this snapshot and RSVP to the event. This will give the event creator a better idea of what kind of interest is out there and how many people may attend. This way they can adjust the meeting location or any food that may have been purchased accordingly. Overall, Owl Snapshots aims to forage a better campus environment for all.

Competitive Analysis

Many different forms of office and campus maintenance software exist. The purpose of this software is to provide residents, employees, and maintenance staff with a unified system to report issues and fixes. Most software also allows the maintenance staff to set routine maintenance and inspection schedules on equipment and buildings. This way many costly issues can be avoided with routine checks. These various programs all have different features or methods of handling these various tasks. Below is a chart of how Owl Snapshots compares to the other products available on the market today.

	Owl Snapshots	Facilities Management Express	emaint	Hippo CMMS	Fastrak Maintenance
Supported OS	Web browser	Mac OS, Web browser	Mac OS, Web browser	Windows, Mac OS, Web browser	Windows, Mac OS, Linux, Web browser
UI	Multifunction tiles	Calendar based	Calendar based, job lists	Calendar, interactive floor plan	Job lists
Automated Routine Work Scheduling	Yes	Yes	Yes	Yes	Yes
Additional Functions	Campus Events	--	--	Barcode support	Inventory management
Maintenance Staff Notifications	Email	Email	App notification	Email, App notification	Email, App notification

As you can see, Owl Snapshots, lacks standard software deployment. This is because we believe that a web browser experience will not only meet your needs, but provide you will a seamless experience. Gone are the days where you have to work with various different types of UI and have to relearn core functions based on the system you are on. Owl Snapshots is designed entirely around the web browser and works the same regardless if you are on Windows, Mac OS, etc. This also means you don't have to worry about keeping our software up to date. When you log in, you'll be working with the latest edition of Owl Snapshots.

Most of the competition relies on a calendar based UI system. While this is a tried and true system, here with Owl Snapshots we believe in innovation. Our tile based UI provided our users with a format that provides information and function in a beautifully designed format.

Like most software solutions you can schedule routine maintenance programs. Such as weekly inspections of critical areas, or servicing equipment. With Owl Snapshots, you can also set up recurring non-maintenance events. Owl Snapshots can be used for more than just allowing users to report issues. Students, faculty members, organizations, companies, etc. can schedule events that will be happening on campus. With maintenance operations and events under the same platform, event planners will know in advance whether or not the location they plan on using will be suitable at that time. Our tile based UI makes all of this a seamless experience.

Lastly our maintenance staff will receive prompt notification from our email service with all the necessary details of the issues at hand. No need to log directly into the service to see the information needed. Everything is provided all at once. After the job is done, the maintenance staff can report the job done, and the person who reported it will be notified. Owl Snapshots is an all in one experience for campus management solutions.

Data Definitions

Users - A person who will use the app to some capacity. Users will have login information and may have varying permissions within the app (For example, "Student" account vs. "Administrator" account).

General User - A user on campus with permissions to create new snapshot posts and add comments or information to existing snapshot posts.

Administrator User - A campus administrator with more control over snapshots and permissions than a student user. They will likely have the ability to completely remove posts once they have been resolved, marked as redundant, or invalid.

Snapshot post- these are the posted issues/events within the app. Each post will contain metadata about the issue/event such as the time of occurence, location, event categorization (Is it critical issue or a casual event?), and perhaps the number of students affected or number of students in attendance of the event.

Use Cases

Owl Snapshots is designed around a functional experience. It is built to be a tool for the maintenance staff as well as everyone else on campus. We can classify the many users of Owl Snapshots into two different types. We have general and administrator. We'll go over how those two different users interact with each other..

We'll start with the most common user, students. They will be general users and one of the main sources of issues reported. Any users familiar with the driving app Waze will understand our layout. When a general user makes a report of a defect, that report will be visible to all users unless otherwise specified. Most of the time, if one student comes across a defect, thousands of other users will as well. To prevent a flood of hundreds of work orders about the same event, users will be able to see the issue has already been reported and can confirm that the issue still exists or provide additional details. Our tile based UI allows users to see all the necessary details to know whether the reported issue is one and the same that they are experiencing.

Use case, "General user would like to report an issue on campus that does not already exist on the web app":

If a student on campus encounters an issue, they can access the web app to report the issue. Once the student has reached the Owl Snapshots, they will need to login with their university credentials to create a snapshot of a new issue. Once the user has logged in, they can create a new snapshot post of the issue. Before creating a new snapshot post, the user may be reminded with a prompt to check that the issue has not already been reported, or a query can be made to show similar issues to prevent redundant snapshots. If the issue has not been posted, the user can create a snapshot of the issue, and will be prompted to enter information about the issue such as location and perhaps severity level.

Use case, "General user would like to report an issue on campus that has already been posted on the web app":

Once a user has logged in and would like to report an issue, they may see that the issue has already been posted on the Owl Snapshots dashboard. They will have the ability to select this issue and confirm whether it still exists or has been resolved (perhaps similar to a Reddit upvote/downvote system), and add any additional comments or information about the issue. It is also possible that a user does not want to look at other issues before posting. To resolve this potential issue, a query can be made based on the issue criteria before it is posted, which would return similar issues to steer the user away from posting a redundant issue or event.

Use case, "General user would like to create a snapshot post of a new non-issue event on campus":

The general process for posting non-issue events will be the same as creating an issue related post. A user will have the option to select whether the post is for an issue or non-issue event when the post is being created.

Use case, “Administrator user would like to review campus snapshot issues”:
Administrator accounts will have special permissions in Owl Snapshots, allowing for more control than the student users. Once an administrator is logged in, they will be able to see the current issues on campus. The ability to sort issues based on different criteria such as most recent or level of severity will be a useful feature for administrators to keep track of issues.

Use case, “Administrator user would like edit the status of an issue”:
Once an administrator is logged in, they may select a snapshot post of an issue. They will have the ability to edit the status of an issue (For example, marking as “Resolved”). Once an issue has been resolved, it may be archived to a list of resolved issues.

Use case, “General user would like to browse events and activities taking place on campus”:
A user will access the web app and will be able to access a list of all events that have been posted on the campus. These events will include information such as time, location, any associated costs, and may allow users to comment and add more information.

High-Level Functional Requirements

1. Users can login to the application.
2. General users have the ability to read all snapshot posts of issues and events, as well as the ability to submit new issues and events.
3. Administrative users should have the ability to also further edit snapshot posts by changing their status or removing posts when needed.
4. Each post contains metadata about the issue or event that can be read by users.
5. Users have the ability to add and read comments for posts.
6. Users have the ability to search for specific events or issues that have been posted.
7. Users have the ability to sort issues and events based on certain criteria.
8. The system has mechanisms in place to prevent or discourage redundant submissions. Such mechanisms may be employed through the use of user-submitted flags to moderate that content is also appropriate and relevant.

Non-Functional Requirements (Quality Attributes)

1. Owl Snapshots will be accessible through student’s MyFAU login credentials and to protect the privacy of student’s they may be allowed to post anonymously.
2. In an effort to maintain accessibility to all users, Owl Snapshots will focus on:
    a. Compatibility with most commonly used web browsers
    b. Image text alternatives (for those vision impaired who use with screen readers)
    c. Sufficient text contrast ratio/resize text settings (greater visibility of text content -- possibly implemented in a different “view mode”)
3. For security purposes, this application should only be accessible to university users with valid login credentials.
4. The backend servers should be provisioned adequately to handle the entire university’s population in the event of a critical issue that would lead to many users at once.

5. A TTL or archival storage solution for resolved issues will help conserve backend memory as the number of posts grows.

High-Level System Architecture

Owl Snapshots will be modeled using an event-driven architecture and mediator topology. User posts qualify as the event, which the Owl Snapshots “administrator” will assess to determine its resolution status.

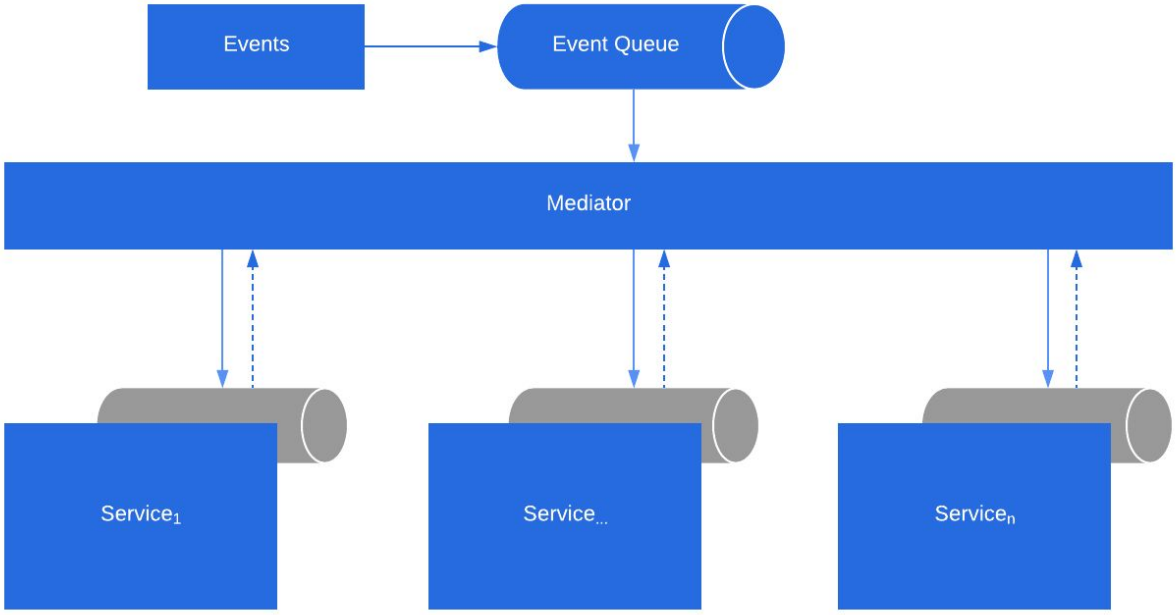

Figure 1 - Mediator Topology

Since Owl Snapshots contains dynamic content, the dashboard experiences constant update from new user events. It will be implemented using Laravel PHP Framework and HTML/CSS for front-end. Owl Snapshots must be able to accommodate a roughly 30,000 student population. However, without an idea of how many clients may use Owl Snapshots the database size (funneled through SQL) and average response time cannot be evaluated at this present time.

Roles

Grant; Scrum Master, Github Master
Kevin; Product Owner, Back-End Lead
Nicole; Scrum Master, Front-End Lead

[**Checklist** status is available as a card on our Trello workspace]