The University of Nevada, Reno

Department of Computer Science and Engineering

CS 426: Senior Projects in Computer Science

Team 05

Aisha Co, Melissa Flores, Joanna Lopez, Nasrin Juana, and Araam Zaremenhjardi

Instructor: David Feil-Seifer, Vihn Le, and, Devin Lee

Advisor: Erin Keith

February 4th, 2022

Table of Contents

Abstract

The current dormitory maintenance requests system at the University of Nevada, Reno (UNR) provides a system that is flawed and susceptible to complications. Some issues the current system faces are paper trails, leading to lost reports, a lack of maintenance request progress tracking for the students, and does not incentivize students to submit issues that occur within the property. Optimum Property Fix (OPF) aims to revamp UNR's current system through a new and improved maintenance request system, accessible to dormitory students as well as facilities and service members. OPF's functionality fully eliminates the paper-based system and replaces it with a web application that retains all maintenance records in one place. Additionally, the project management of OPF entails the assessments of the project's progress and its risks, legal and ethical aspects, identification of the application technology stack, and organization of team members to fully accomplish the objective.

Project Description

The main goal of the project is to eliminate the paper trail currently used at the University of Nevada, Reno between dormitory students and Facilities Services using a web application. The intended users of Optimum Property Fix are dormitory students, residential assistants, university administrators, and facilities & maintenance workers at the University of Nevada, Reno. The project will be beneficial to dormitory students to better facilitate communication between themselves and facilities and maintenance. Dormitory students will also benefit from electronic ticket requests making them accessible via the internet compared to the current system of maintenance requests in paper format. The University of Nevada, Reno administration will benefit from OPF by being able to view real-time analytics relating to building health and preventative maintenance. Additionally, administrative staff will be able to view billable hours of maintenance workers and inventory spending of each dormitory. The facilities and maintenance workers will be able to benefit from OPF by viewing maintenance requests online. Moreover, the application will aid workers by preventing loss of paperwork, tracking of maintenance requests, and ultimately being able to schedule maintenance. The public interest in the product would be the parents of dormitory students who are paying the University of Nevada, Reno housing. Additionally, caring for the safety of students and the welfare of the Reno community.

The main functionality and capabilities of OPF are handling of the maintenance requests between its users such as the creation and visualization of maintenance requests. In addition, communication between the students and administrators such as feedback, frequently asked questions page, and communication center is available to the users. Finally, a chatbot and an analytics page containing data about each dormitory are part of the functionality and capabilities of OPF.

The back-end technologies for the application are a Python Flask framework and MySQL for queries. The front-end technology includes React using HTML, CSS, and JavaScript. The libraries used for the project can be viewed in Appendix A1. No external devices are needed except for a mobile device with an internet connection. OPF currently has no hardware components.

The dependable property regarding reliability is the ability of the system to deliver services as specified [4]. The way to achieve reliability in our web application is to avoid errors when developing the application. This can also be achieved through UX/ UI design and meeting users' expectations. Next, regarding security is the ability of the system to protect against unauthorized or deliberate intrusion [4]. The way to achieve security in our web application is to have a layer of protection to shield against external attacks. This can be applied to our user login system and prevent access to the web application database. Safety is the ability of the system to operate without fatal failure [4]. The way to achieve safety in our web application is to include recovery mechanisms to help restore service after failure. Additionally, this can be accomplished by creating a website application that can resist accidents and/ or intrusions.

Significance

Optimum Property Fix (OPF) is a web application designed to assist the University of Nevada, Reno (UNR) dormitory facility services regarding maintenance requests. The current system has paper trails, is vulnerable to being lost, does not incentivize submission of minor issues within living spaces, and lacks progress tracking of maintenance requests to students.

OPF will rectify issues from the current system while serving the same functionality of creating maintenance request tickets from the dormitory residences to the Facilities Services. Additional functionalities would be the ability to visually keep track of their request tickets, allowing students to identify the status of their tickets and Facilities Services to organize. Furthermore, students can provide feedback for the tickets, allowing communication between the two parties regarding the maintenance requests. Similarly, there is a frequently asked questions (FAQ) page for residences to access. There is a chatbot for guided page navigation. Specifically for the admin users, there is an analytics page containing data about each dormitory, allowing each building's health to be monitored. Additionally, administrators will be able to view preventive maintenance of buildings.

This project allows the further development of teamwork, multi-tasking, and time management skills. Each member has to manage their work and personal schedules. By balancing their schedules while working on the project for an extended period of time, the members would be able to develop skills that would be useful for future projects and industry.

Related Projects and Market Potential

There are similar related products on the market such as the Fiix [1], NET Facilities [2], and Maintenance Care [3] with different features applicable to Facilities Services. Figure 1, is the competitive outline of each of the three related products alongside OPF and the different integrations on their unique platform.

		Competitive Analysis			
		OPF	**Fiix**	**NET Facilities**	**Maintenance Care**
PROFILE	**OVERVIEW**	OPF web application lets one manage student housing maintenance.	Fiix Software for school maintenance lets one manage maintenance for affiliate sites abroad, buildings on campus, outdoor fields, and everything in between. Full access to powerful tools that will help you simplify maintenance management strategy.	NET Facilities links together and manages administrative buildings, libraries, classrooms, student housing (dormitory), modular buildings, grounds, athletic departments/ gymnasiums, auditoriums, fleets, campus medical facilities, faculty centers, and others.	Maintenance care can help the maintenance team track work orders and PPE, start preventative maintenance programs, develop asset tracking, track progress, and report on progress. Students, teachers, and administrative staff need a functional environment to succeed; school management software can help them be the best leaders for our youth.
	COMPETITIVE ADVANTAGE	Customization – Built inclusively for the UNR.	AI-powered reports.	Used in the vast majority of industries such as education, government, property management, health care, etc.	Custom Built Features.

MARKETING PROFILE	**TARGET MARKET**	UNR	All industries	All industries	All industries
	MARKETING STRATEGIES	None	Social Media, Free Online Demonstrations	Social Media, Free Online Demonstrations	Social Media, Free Online Demonstrations
PRODUCT PROFILE	**PRODUCTS & SERVICES**	Maintenance requests, Preventative Maintenance, Reports/ Metrics, Task Management, and Help Center for Students.	Training, and Implementation, Help center, and Fiix Community. Asset, Parts & Supplies, Work Order Management.	Work Order Management, Asset Management, Preventative Maintenance, Inventory Management, and Key/Lock Management.	Work Order Management, Task Management, Preventative Maintenance, PPE/ Tool tracking, Reports, and Alexa integration.
	PRICING & COSTS	N/A	Quote: Custom Pricing (Pay-as-you-go)	Quote: Custom Pricing (Pay-as-you-go)	Quote: Custom Pricing (Pay-as-you-go)
	DISTRIBUTION CHANNELS	Direct to Consumer	Direct to Consumer	Direct to Consumer	Direct to Consumer
SWOT ANALYSIS	**STRENGTHS**	Customization	AI integrated Software	Already in most industries	Custom to customer needs
	WEAKNESSES	No actual product built, features not decided.	Not local, time zone difference, and no 24-7 customer support.	Not customizable to fit all needs and wants.	Too slow to adapt to change i.e., newer technologies

	OPPORTUNITIES	Potential for usability and streamlined process for students and Facilities & Management.	Improving customer satisfaction and industry awards.	Building strong customer relationships	Improve overall product quality design and product features
	THREATS	Not creating products in time for student use	Possible new product failures	Maybe overly price compared to most competitors	Products becoming too outdated

Fig. 1: Market Analysis of OPF Competitors.

OPF's market potential depends on the student's behaviors which affect the state of UNR's properties, competition, and consumer patterns. The application offers optimum service, assistance, and accessibility; the expected market potential is promising. OPF ensures to offer fast and proper service as maintenance issues cause inconveniences, especially to the already busy college student. The design of the OPF web application and its implemented technologies will help dormitory students find the appropriate service for their needs, leaving them to always turn to OPF.

Social and Environmental Impacts

OPF's purpose and integration will allow for beneficial social and environmental impacts starting with replacing the current system of the university. Getting rid of paper trails and replacing them with OPF, will not only solve lost reports but will also reduce paper waste, keeping the best interest in mind for the environment. In addition, creating and receiving maintenance requests will be easier to access for both students and the facility, leading to more awareness of any issues that arise in the buildings. As more problems are identified and reported, better and continuous maintenance of buildings means ensuring the safety of students.

Future Developments

After creating the final version of OPF, future features can be implemented. OPF can contain artificial intelligence to organize the maintenance tickets based on priority and severity. A messages system can be implemented, so there is direct communication between students and Facilities Services. Lastly, if OPF becomes normalized, OPF can become a campus application instead of a dormitory-only application since all buildings require maintenance requests.

Legal and Ethical Aspects

The Legal and Ethical Aspects of the application involve multiple parties such as the university, facility services, and the dormitory students of the university that expect ethical and functional software with ACM's Code of Ethics Product Clause as a guiding principle. OPF's legality implications are within the realm of possible contracts the university may have with a third-party company to upkeep the application with associated costs while striving to have good software quality. Section 3.0.1 of ACM's Code of Ethics describes the need for software to be affordable while it still retains objectives described by stakeholders. The university and facility services expect OPF to be reliable and accessible to all dormitory students. This means each party involved (being the university, facility services, and students) has a stake to ensure OPF meets good software quality. The university's stake in OPF is ensuring the housing department is able to function effectively using the software without major complaints from the public or major budget deficits due to OPF not being able to accomplish its objective of being a one-stop shop of building maintenance. The facility service's stake in OPF is being able to upkeep dormitory buildings from issues as outlined in the ACM's Code of Ethics - Product Clause of Section 3.01. Good software quality is ensuring the application and its related modifications remain active and free of possible exploits that may compromise the system. The team's plan to ensure that the product meets the highest professional standards possible is to establish achievable goals and milestones set for any modifications and features wished to be proposed. To avoid risks of legal and ethical issues, proper testing, debugging and documentation are conducted to verify professional, high-quality software. The specifications of the software make certain of the user requirements proposed by OPF.

Changes and Progress since the Initial Project Concept

The progress made towards OPF would be the implementation of the student pages and the chatbot based on the user interface (UI) design developed through Adobe XD. The design follows the initial layout of the UI design as well as the color scheme to make the web application clear and uniform. Correspondingly, this allows the users to intuitively navigate OPF. The features that are developed include the user sign-in page, create new maintenance request page, dashboard page, view maintenance requests page, and feedback page for the students and

the chatbot. The integration of the student webpage version and the chatbot creates the experience of navigating through the OPF web application. Currently, the web pages and the chatbot allow general interaction between the web application and the user.

Changes from the initial project concept include the reformatting of the communication center between the students and the administration. The communication page will be transformed into forms that students will be able to submit and post on a dedicated page. The administrator will then be able to reply to the questions and comments later and these answers will be publicly viewed by everyone using the application. If the students have any private questions, they can email the administrators and communicate with them. This will make it easier for users and any students with similar questions will be able to gain answers from the questions.

In addition, OPF will have an analytics section. The analytics section will compromise the statistics about the request ticket, buildings, and inventory. The users will be able to view statistics regarding the inventory for each building in order to ensure preventive care for the buildings and monitor the conditions of the buildings and their corresponding components.

Finally, the team has decided on the removal of the AWS and DynamoDB for the back-end technologies and AI recognition for objects and images. AWS and DynamoDB will be replaced with Flask and SQLite. The reason for these changes is that DynamoDB is a non-relational database, which is not suitable for the OPF application. As such, SQLite, a relational database will be utilized for the database. Additionally, AWS, which was selected because of its useful functionalities with AI recognitions and available memory for the database, will be replaced with Flask. Right now, the team has a small amount of data that they are able to test and store in their local machines. In addition, the team's main focus is building the components for the maintenance requests and the analytics within the semester time frame. As such, Flask will be utilized for the back-end framework and to create the log-in page to make the implementation easy and fast for the team. Once the team is able to accomplish these goals and has time remaining then, they will implement AI recognitions for Images and Objects and connect and store the database in AWS.

Project Responsibilities

OPF's architecture involves two main subsystems being the frontend and backend of the application that have their separate subsystems that team members are responsible for ensuring proper functionality. The two subsystems communicate with each other through the use of a JSON-based REST API with separation of concerns of user views, error handling, and data processing delegated between the two systems. The front end of OPF's technology stack is made up of standard web technologies such as JavaScript, Cascading Style Sheets (CSS), HTML, and libraries such as React, React Router, and other React libraries. The backend of OPF's technology stack is made up of Flask as a basis for the backend, Flask-SQLAlchemy for ORM

support, SQLite database, and additional libraries denoted in the Appendix. The frontend is composed of several subsystems including a component used to create consistent input elements within the application, the router is used to switch web views, the authentication system for session functionality functions used to conduct API calls to the backend for data processing to the database or retrieval from the database, and additional libraries denoted in the Appendix. The backend is composed of several systems of error handling for disputed data, routing to create endpoints for different REST API calls, access token management queries built using the ORM, and the database itself for data persistence.

Araam Responsible for Routing and REST API Calls

Araam is responsible for the planning of frontend and backend routing of OPF and functionality for communication between the frontend and backend through API calls. Planning of the frontend routing involves ensuring state data such as user management, session tokens, and other data required for the application to effectively operate are correctly passing data between different web views through "stateful" components. The primary link between the backend of the application and the frontend is the REST API being used as an interface to send and retrieve data and queries. Araam is responsible for defining a consistent specification to structure JSON objects that are both generated and consumed by the frontend and backend server and a consistent method is used to respond to errors that may originate from users incorrectly inputting data into the application. Araam is tasked with properly structuring the backend routing of the application to allow the backend to expose points where data is sent through the REST API from the backend and lay the groundwork for data to be properly vetted for the database transactions. In conclusion, Araam is responsible for ensuring the frontend is properly routed for different functionality of the application and developing a framework for communication between the frontend and exposed points in the backend through the use of a REST API.

Melissa Responsible for Maintenance Requests and Frontend User Experience

Melissa is responsible for the overall user experience of the frontend and ensuring OPF's primary features are able to respond and interact with user input. The subsystems for the frontend are the feedback section allowing dormitory students to send and receive feedback based upon their maintenance request experience, dashboard experience, and OPF's primary feature of viewing, managing, and creating maintenance requests. Melissa is responsible for making these subsystems "stateful" by allowing data to be dynamic using the React library. Components within these subsystems will be able to pass data from form elements such as buttons and text areas and forward them to either different areas of the application or the backend. This area of the frontend design will be in collaboration with Araam. The frontend is divided between two user experiences being the administrator experience (for facility services staff) and the student experience. Melissa is responsible to ensure both these divided experiences are able to have a consistent flow and experience for users while still retaining unique functionality and styles.

Maintaining a refined user experience involves proper structuring of application pages and defining of CSS styles used within the application. Therefore, Melissa is responsible for a consistent experience across the application student and administration views while making these experiences dynamic based upon user interaction.

Joanna Responsible for Backend

Joanna is responsible for creating necessary queries to answer user queries and in collaboration with Araam developing necessary backend routing to receive and send query results to the frontend. Tasks associated with creating the necessary queries are being able to analyze database tables and using the Flask-SQLAlchemy ORM to programmatically fulfill queries in viewing tickets, feedback, user information, joining tables to formulate complex queries based upon database models, and then be able to send the results to the database. Backend design by Joanna will be formulating a scalable specification for creating and calling user queries and being able to create serializers for queries to be converted to JSON objects. Joanna is responsible for assisting Araam with the user queries and using serialized database objects for the backend's REST API endpoints. Overall, Joanna is responsible for developing user queries that are serializable in that are to be communicated to the frontend through the developed REST API to further foster an interactive experience for users using the frontend subsystem of the application.

Nasrin Responsible for Database

Nasrin is responsible for creating database models for OPF's SQLite database. Using the Flask-SQLAlchemy ORM, Nasrin will use the library to programmatically define with Python necessary database models and attributes related to the defined database models. In collaboration with Joanna, the database models will feature relationship mapping using primary and foreign keys used to abstract relationships of the database tables. Nasrin's responsibility is to overall maintain the SQLite database with its models and ensure the database is flexible to satisfy simple to advanced user queries.

Aisha Responsible for Analytics

Aisha is responsible for the Analytics functionality of the application. The Analytics functionality is composed of two subsystems being the frontend view and the logic within the backend processing data from the database to make useful analytics. Implementation of graphical charts will involve user experience design from Melissa and ensuring analytical data is available only to administrator users will involve establishing proper routing in collaboration with Araam. The top analytics metrics to be implemented are the number of tickets generated by building and having the ability to view the type of maintenance requests through bar graph visual representations. Actual data processing will require implementation within the backend using Python with formatting the data to be properly handled by the REST API so that the React frontend will be able to properly render analytical data. Aisha is responsible for ensuring

administration users are able to properly request different visual representations of collected data from maintenance requests and make useful metrics that further enhance the productivity of facility service staff.

Project Monitoring and Risks

The team has developed an outline to ensure that the project is continuously progressing to eventually meet the completion and fulfillment of goals. First, the end goal is defined and the team has an ideal vision in mind. The overall mission is understood before diving into the tasks required to complete it. With this, weekly meetings are held within the team to work on project assignments, discuss what has been accomplished so far, and contribute to the components of the web application. In addition, meetings and discussions with the team advisor are required to make sure the team is guided on the right path and to gain new perspectives. Productivity, communication, and accountability are values that the team upholds to make certain the expectations are met. Achievable and ambitious milestones are set in the degree of long-term and short-term, so it is important to delegate tasks and distribute a balanced workload within the team of 5 so that timely deliverables are accomplished.

With big and ambitious projects such as Optimum Property Fix (OPF), come many risks to consider. The team has identified these risks and developed strategies to mitigate them. The first risk the team might encounter is the communication of the front-end part of the application with the back-end part. The strategy applied to this risk is to seek help and guidance from the team advisor. With this, a goal is well-defined, small tasks and expectations are then set on a weekly basis, allowing the team to tackle and resolve this risk in small steps. Next, a risk may come from the user log-in page not operating as expected. To mitigate this, the team plans to highlight and define the scope of this area in the application and as well as develop workshops and meetings that target this specific concern. Scheduling risks may occur during this period of completing the project and meeting certain deadlines. This risk includes delays in deliverables, failure to meet deadlines and expectations during demos. A solution to this would be to plan regular meetings with the team and advisor. In these meetings, discussions will be productive and planning will be efficient. In addition, a review of all tasks is important to sort in priority level and tackle risky activities earlier. The team may lose control over web application feature priorities leading to the inability to fulfill important features. To mitigate, the project sponsor will make clear the importance of these features and how it will add value to the application. Next, final dates and rollouts will be communicated as well as coming up with alternatives to ensure that there is a backup plan. In addition to programming and learning new software, there is a risk that the front-end and back-end technologies we use do not work together. Specifically, with using React and Flask, there is a concern that the React server may fail to communicate with Flask. As the team is split up between front-end and back-end, it is important that there is clear communication between the two and set expectations between the two teams because one team

needs to function for the other team to work. As a part of scheduling risks, the team will encounter changes in schedules and deadlines, leading to difficulty in work estimation. A team sponsor will be responsible for monitoring weekly schedules and include reviews as a part of the agenda items. Lastly, communication may be hard to manage leading to a lack of clarity of certain tasks and confusion. It is important that each team member attends the majority of team and advisor meetings as important discussions occur. Setting clear boundaries of when and where to communicate about the project as well as knowing when it is important to take breaks to prevent burnout. Lastly, asking questions may solve the majority of confusion that arise as well as improve the team dynamic. Figure 3 displays the OPF Risk Register that is used to identify potential risks in the project that can ultimately derail intended outcomes and figure 3 displays the key utilized to structure the OPF Risk Register for the 'Likelihood' versus 'Impact'.

		Likelihood				
		1	2	3	4	5
Impact	1	Low	Low	Low	Medium	Medium
	2	Low	Medium	Medium	High	High
	3	Low	Medium	High	High	Extreme
	4	Medium	High	High	Extreme	Extreme
	5	Medium	High	Extreme	Extreme	Extreme

Fig. 2: The key used to structure the OPF Risk Register for the 'Likelihood' versus 'Impact' which ranks each category: one to five and describes the severity by low, medium, and high keywords.

Risk Register

Project Name: Optimum Property Fix

			Current Risk								Risidual Risk		
Risk ID	**Date raised**	**Risk Description**	**Likelihood**	**Impact**	**Severity**	**Owner**	**Mitigating Action/Strategy**	**Contingent Action**	**Progress on Actions**	**Status**	**Likelihood**	**Impact**	**Severity**
1	1/25/22	Front End not talking to the backend.	Medium = 5	Medium = 5	Extreme = 25	Project Sponsor: Joanna Lopez	Complete a plan of action with Advisor: Erin Keith. Provided a well defined goal to accomplish small tasks for weekly updates.	Escalate to the CS 426 Team for external help and raise concerns during project demonstration. Assess the risk of uncompleted project.	Created database tables and start testing the tables.	Open	4	5	20
2	1/30/22	User login not completed or functional.	Medium = 4	Medium = 4	Extreme = 16	Project Sponsor: Araam Zaremenhjardi	Define the scope in detail via design meetings with input from Advisor: Erin Keith.	Document assumptions made and associated risks with login page. Request high risk items that are not defined removed from scope of the project.	Design workshops/meeting scheduled with Advisor.	Open	3	4	12
3	1/25/22	Project schedule is not clearly defined or understood	Medium = 4	Medium = 4	Extreme = 16	Project Sponsor: Nasrin Juana	Hold scheduling meetings with the Team to help understand the plan of missed tasks.	Share the plan with the Team via Slack and go through upcoming tasks at each weekly progress meeting.	Meetings scheduled.	Open	3	4	12
4	1/25/22	No control over web application feature priorities.	Low = 3	Medium = 4	High = 12	Project Sponsor-Melissa Flores	Each Project Sponsor will brief Team on the importance of their project features. Will communicate final dates/ roll outs after the scheduling of meetings. Identify backup plans and resources on the project.	Escalate to the individual Project Sponsor and help with back up resources.	Individual Project Sponsor has agreed to brief Team in meetings.	Open	2	3	6
5	2/2/22	React server to Flask communication	Low = 3	Medium = 4	High = 12	Project Sponsor: Araam, Aisha, and Melissa	Communicate tasks early. Ask again if Front-End team needs anything from Back-End Team.	Escalate to Project Sponsor and Advisor for help.	One feature fully functional.	Open	2	3	6
6	1/25/22	Estimating and/or scheduling error for application.	Low = 3	Low = 3	High = 9	Project Sponsor: Araam, Aisha, and Melissa	Use methods of work estimation, and carefully track and forecast project completion and making adjustments as necessary. Build in a contingency plan and scheduling. Track schedules weekly and include review as an agenda item in every project team meeting.	Escalate to Project Sponsor and Slack. Raise request for change to schedule.	Contingency agreed by the Team with guidance of Advisor.	Open	2	2	4
7	1/25/22	Unplanned work that must be accommodated	Low = 2	Low = 3	Medium = 6	Team Manager-Araam	Attend all Team meetings and Advisor meetings. Check previous work and team updates. Check all plans and document all assumptions made in planning and communicate to the Team.	Escalate to the Project Sponsor and Advisor with plan of action, including impact on time, and quality of work.	Attending scheduling workshops.	Open	1	2	2
8	1/25/22	Lack of communication, causing lack of clarity and confusion.	Low = 2	Low = 2	Medium = 4	Project Sponsor: Aisha and Nasrin	Writing a communication plan which includes frequency, and goal of each communication. Using the most appropriate channel of communication for members i.e. Slack versus immediate response via text.	Correct misunderstandings immediately. Clarify areas that are unclear to individuals or Team by asking Project Sponsor for clarification.	Communication plan in progress.	Open	1	1	1

Fig. 3: The table describes the OPF Risk Register that is used to identify potential risks in the project that can ultimately derail intended outcomes. This tool includes information such as the level of risks, who owns the item, the date raised, and the mitigation measures in place to respond to each risk. Ultimately, the described Risk Register will aid in project management and help facilitate communication with stakeholders and the project team.

Contributions of Team Members

Araam Zaremehrjardi's Contribution

Araam Zaremehrjardi's total time worked on the revised concepts and project management totals five hours. The contributions include assisting in writing the Legal and Ethical Aspects section and writing the Project Responsibilities section. The Legal and Ethical Aspects section was written in conjunction with Melissa Flores in developing an outline for the section and further developing the outline into the section. The Project Responsibilities section was in total outlined and developed into a section by Araam.

Joanna Lopez's Contribution

Joanna Lopez's total time worked on the revised concepts and project management totals four hours. The contributions include working on the project description and the Appendix for the libraries and tools used to create OPF. In collaboration with Mellissa Flores, the 'Risk Register' and the key were created.

Melissa Flores' Contribution

Melissa Flores's total time worked on the revised concepts and project management totals 4 hours. The contributions include working on the Legal and Ethical Aspects section with Araam Zaremehrjardi by outlining and looking into the ACM code of ethics. In addition, contributions included writing the Project Monitoring and Risks section along with Joanna Lopez who developed the risk register. Lastly, more contributions were to assist Aisha Co in writing and editing the project's social and environmental impacts section.

Nasrin Juana's Contribution

Nasrin Juana's total time worked on the revised concepts and project management totals four hours. The contributions include the writing of the changes and progress since the initial project concept section. In addition, overlooking and editing the entire document once each team member has completed their section.

Aisha Co's Contribution

Aisha Co's total time worked on the revised concepts and project management totals to three hours. The contributions include writing the project significance - its importance, its market potential, and its further potential after the semester is over. Similarly, worked in conjunction with Melissa Flores regarding editing the project's social and environmental impacts.

References

1. "The #1 School Facilities Maintenance Software Platform." *Fiix*, 23 Aug. 2021, https://www.fiixsoftware.com/cmms/industry-solutions/university-maintenance-software/ .
2. "CMMS Software for Maintenance, Facility and Property Management." *NetFacilities*, https://www.netfacilities.com/?hsLang=en.
3. "Management Software Designed to Simplify Maintenance at Schools." *Maintenance Care*, 13 May 2021, https://www.maintenancecare.com/education.
4. "Software Engineering System Dependability." *CS 410/510 - Software Engineering Class Notes*, https://cs.ccsu.edu/~stan/classes/CS410/Notes16/10-SystemDependability.html.

Appendix A-1

Libraries and Tools

Front-End

1. "Apexcharts" version: "^3.32.1"
2. "React" version: "^17.0.2"
3. "React-Apexcharts" version: "^1.3.9"
4. "React-Calendar" version: "^3.5.0"
5. "React-Dom" version: "^17.0.2"
6. "React-Router-Dom" version: "^6.2.1"
7. "React-Select" version: "^5.2.1"
8. "React-Simple-Chatbot" version: "^0.5.0"
9. "React-table" version: "^7.7.0"

Back-End

10. "Flask" version: "^2.0.2"