DATE: December 2, 2022
TO: Dr. Hu Tao
FROM: Joshua Pacis, Nick Daarol, Joshua Ruiz, Eduardo Ochoa Aragon, Casey Kaufman, Mkrtich Taymizyan
SUBJECT: Project Progress Report - Milestone #3

With the creation of a new system that will inform engineers of a machine that is having a problem, it is time to move to the next step which is to implement our created system to Round 1's current system. Our system will be added as a tab on the bottom of the page where various links are such as rules, contact information and various other informational links. The QR codes that we scan will take people to this tab which as mentioned before allows people to select a machine, the problem they have and other information they would like to add. We also developed a maintenance plan that details how maintenance will be performed as well as when it will be done.

IS Implementation

The Information System Project will be implemented with the assistance of Cloud Computing Services in either Amazon Web Services (AWS) or Microsoft Azure. Either of the two will suffice, however, we recognize the potential benefits in choosing one over the other. Microsoft Azure generally seems to be the easier of the two for development teams to familiarize with, allowing for easier and more efficient implementation. AWS, however, seems to be much more flexible and cheaper overall, at the cost of having a higher learning curve, which could cause schedule delays. Regardless of which service platform the team decides to opt for, the implementation of the project will use the services these technologies provide to code, test, and implement our proposed information system. Since these technologies are cloud services, there is no need to install them on a physical device. All that is required is an internet connection.

Coding

As far as basic coding goes, the team foresees two main components to developing and implementing the proposed system: a database and a system that captures user input from the website interface and assembles it into an entry in the said database.

Through the services offered by Microsoft Azure or AWS, a MySQL database will be created and hosted. The database itself will utilize Data Definition Language (DDL) to create a collection of logically related tables. The system to capture user input will be directly integrated into the web interface. The web interface will utilize HTML language to program the interface to capture the inputs from the users and assemble them into entries to be recorded into the MySQL database.

Since the design of our web interface mainly makes use of drop-down lists, all that is necessary is that the coded program capture which entries from the drop-down list were selected, and store them in the appropriate tables and cell entries of the database. Documentation of each module, each part of the code, each resource used, will be recorded and kept, of course, to understand the implementation of the project even after several new iterations of the information system. Alongside the actual code, documentation will serve as the main deliverables of this phase.

Testing

The information system's implementation will be tested via rigorous "desk checking" and "walkthrough" methods. Once coding has finished, testing will start immediately with an internal walkthrough conducted by the staff team who coded the system. A general "walkthrough" test will be carried out as follows:

- Simulate customer interaction with the website user interface.
- Manually enter "dummy" input to capture information from.
- System captures input information to generate a request ticket, the details of which are stored in the MySQL database.

Following a successful internal walkthrough/desk check, testing will be transferred to regular employees (supervised by the coding team) to perform desk checks. The general process is the same as a walkthrough. If both tests conclude with favorable results, the project implementation will proceed. Testing documentation will also be recorded and kept, as with coding.

Installation

User guides and a training plan will be necessary in order for the employees to properly operate the repair kiosk system when it is implemented. The user guides will cover how to navigate the back-end of the incoming request and harness the information that is collated. System documentation will provide information on the design specifications and functionalities of the service request kiosk. Thus, it will include a breakdown of the search functionality that repair techs will have access to based on the data that comes in from repairs being requested. In-house repair technicians will be solely responsible for the software setup of the kiosk. The user training plan will teach employees how to resolve ticket requests with the new system to carry out arcade machine issues in a timely manner. Scenario Based Training will be used as current repair techs are knowledgeable on the overall hardware, firmware and software of our arcade machines. This gives primary users (repair technicians) a first hand interaction with the UI on the back end of incoming repair requests. With the new integration of our online repair kiosk request system, the repair staff will be given scenarios where repair requests will be submitted and will be pending review. Repair techs will then see the log of repair tickets and learn how to identify and interact with the layout of the UI to determine the severity of the needed repairs and how to document their progress. Having an overall understanding of the system in place and its intended use of the information system will be vital in effectively sorting through requests, determining which machines require further inspection, documenting repairs, ordering parts if needed, scheduling repair work, and indicating completion dates and resolving ticket requests from the database upon the completed repair or maintenance of a machine.

The information system will be directly installed to the premises, as, based on Round1's current business model, there is no current IS System in place that provides the functionality that the newly implemented system provides. Effectively, the new system will immediately replace the current system that was being used; that is there is none. Physical installations will be carried out by repair technicians to install a single kiosk that is designated to each Round1 storefront. By design, the kiosk is a simple yet effective way Round1 customer(s) facilitate the initiation of front-end repair requests that can be collated and organized into back-end infrastructure, allowing for efficient and organized execution of operations to repair an arcade machine in question – all leading back to the customer(s) using the machine satisfied.

Barring the database and web interface, there will be no software to install, save for a simple web browser installation on the actual kiosk. The web browser in question will be modified, installed, and maintained by Round1 repair technicians. The modifications to the web browser involve locking or limiting browsing to just the Round1 repair ticket interface.

Directing customers to the online request portal and generating repair tickets is the only functionality the kiosk provides, so any maintenance needed will be limited to browser updates and functionality augments, all of which will be covered in the "Maintenance" section. The kiosks are expected to use the same terminals as the current Round1 POS systems, so the decision of updating the hardware for the repair ticket kiosks will be left to the discretion of Round1.

While most Round1 storefronts have large floor space, machines are irregularly moved around to accommodate new machines, better organization plans, etc. Thus, we will leave the final placement of the repair ticket kiosk to each Round1 storefront's discretion. We suggest the following locations, but to reiterate, each Round1 storefront will handle the final decision.

- By the Front Desk, near the card recharge machines
- Near the Prize Exchange area
- Near the most frequented card recharge machines

Although the implementation of the software and interface heavily resembles that of the cashier and bowling departments, we have concluded it may need further explanation and/or training to successfully serve and collect information the way we intended it to. To address this issue, we have set up a 2 to 3-hour training session via an online video course that will require all managers to have it completed at least a week before deployment of the cabinets in that specific location. The course will provide small questionnaires at the end of each video to make sure a standard comprehension of the software and hardware is met. After training, each manager can use discretion to decide which employees, aside from technicians, should undergo the same training and be in charge of managing repair requests recorded in the system, as well. Further questions and issues will be redirected directly to the manager, who will also be provided with a set of troubleshooting manuals and manufacturers' phone numbers in case of unexpected malfunctions.

Each manual has common content guidance for each cabinet and provides a series of steps in order to troubleshoot and manage the various machines. With these tools and knowledge that we have provided, we rest assured that each location should be fully prepared to implement these cabinets and have the software working properly within the next couple of days.

IS Maintenance

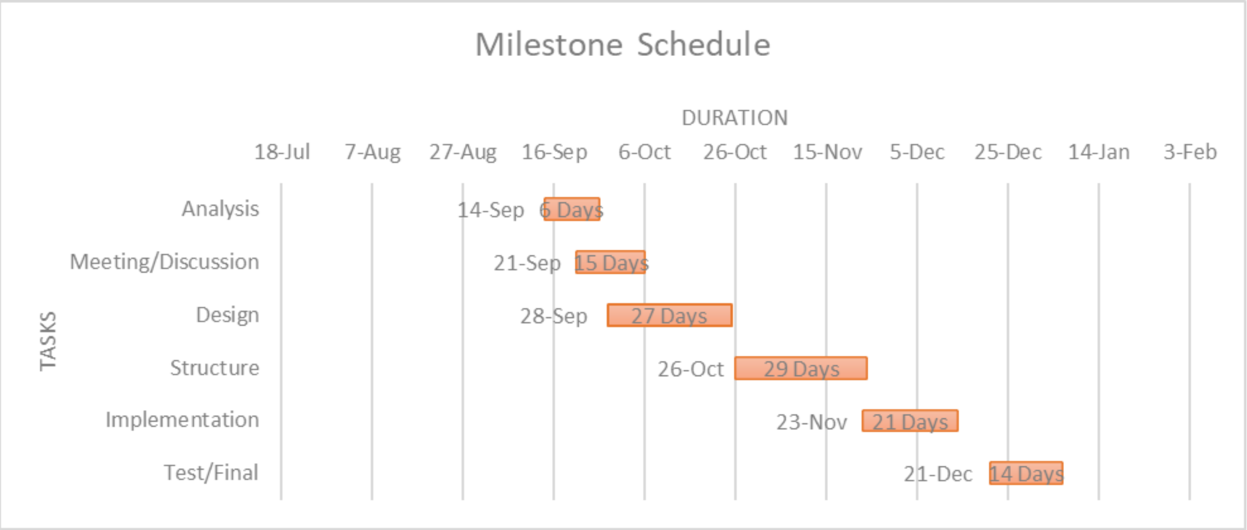
Unfortunately, if there is an issue with the machine, we have no "special" way of maintaining the system. When we are alerted to a problem with any part of the system, maintenance will be as simple as freezing the entire system and manually testing every potential user input action to see where the error originates from. This will typically take place outside of normal business hours, but may also be performed during normal business hours if absolutely necessary. Freezing the system and applying changes will be done through the use of a Configuration Management Tool (CPM). The team has not displayed any preference for any specific tool thus far, so the specific CPM brand will be Red Hat's Ansible. Finally, to document the system and any changes to it, the use of a reverse engineering tool (automated development tool) will become regular practice during any and all maintenance activities. Once again, the design team has not expressed any specific reverse engineering tool, so they tentatively opt for something like Microsoft Visual Studio.NET or OllyDbg.

We expect the types of maintenance on our proposed information system to fall under a mix of Corrective, Perfective, and Preventive maintenance. Corrective and Preventive Maintenance regarding our proposed IS solution is self explanatory – we expect there to be bugs, glitches, or unintended errors that require remediation, as well as some-way to protect and prevent against such errors. Perfective Maintenance mainly pertains to the User Interface portion of the system. For certain, the main system could be improved in some way, however we struggle to produce any impactful improvements. The user interface, on the other hand, could most certainly be improved both in functionality and appearance. When we were deciding on a design for the User Interface, we settled on the mockup with the less appealing appearance, not because it prioritizes functionality at the cost of appearance or anything of the sort, but because it would require less work and time to implement. Through Perfective Maintenance, however, we may yet implement the mockup that was both functional and pleasing to look at.

At this stage, the cost of maintenance of maintenance is still fairly difficult to calculate, given that the system's practical value and necessity (value and necessity determined when the system is officially deployed and made publicly available) has yet to be seen, potentially affecting maintainability. Theoretically, we believe the system will benefit the company, however theory can often differ from real world scenarios. As such, the figure we arrive at is based primarily on assumptions. We expect maintenance to greatly depend on the number and quality of programmers assigned to maintenance. If we assume the maintenance team will also be the development team, then we can forgo the costs of hiring new programmers and simply add to their current salaries. For that we can reserve at least upwards of (tentative) $15,000 for maintenance personnel and any training required to make them familiar with the system, though that number can rise to a higher figure if a bigger maintenance team is desired. There's also the expenses for the video course and technical solution manuals for each machine. Round1 has a great variety of machines, and as such, should assume an amount that rises with each unique machine present at Round1.

Lastly, we consider the cost to maintain the Round1 website. addition should be added to the overall cost to maintain the Round 1 website. According to a few sources, website maintenance can cost around $60-$60,000 a year, depending on operations and severity of maintenance. Those two figures bring the (tentative) total cost of maintenance of our Internet System to be around $75,000 or greater.

Milestone Schedule



Project Status

Task	Scheduled	Projected	Actual Complet	Status	Issues
Analysis	Week	8 Days	6 Days	Compl	N/A
Meeting/	Week	14 Days	15 Days	Compl	N/A
Design	Week 4	28 Days	27 Days	Completed	Re-designed/
Structure	Week	29 Days	29 Days	Compl	N/A
Implementati	Week	31 Days	21 Days	Compl	N/A
Test/Final	Week 16-24	21 Days	14 Days	Completed	N/A

Work Completed, Remaining and Current

- Proposal and Planning
- System Analysis/Feasibility Studies
- Development
- Implementation
- Testing

The above is a checklist of tasks that have yet to be implemented. All stages have been completed. See the "Shutdown" regarding further information on future tasks.

Shutdown

Now that testing and implementation has concluded, the project is now primed to begin shutdown of project planning and development and enter the maintenance cycle.

Official and complete shutdown of the project will be determined by the need of the proposed system in Round1's current business model. The project will be shut down if Round1 does not see frequent use out of the system. Shutdown is expected to involve the following activities:

- Take down the web interface of repair ticket system from the Round1 website.
- Remove repair ticket kiosks from Round1 storefronts.
- Archive and/or liquidate any data relating to the project from the central database.

Conclusion

Implementation and testing of the proposed information system is (tentatively) complete. The proposed information system will now enter the semi-indefinite cycle of maintenance. Hopefully Round1 Bowling and Amusement finds great use in our proposed information system.

References

J., T. (2022, December 5). *Website maintenance cost in 2022: A full pricing breakdown*. Hostinger Tutorials. Retrieved December 11, 2022, from https://www.hostinger.co.uk/tutorials/website-maintenance-cost

PRESIDENT, B. C. W. E. B. F. X. (2022, December 8). *2022 website maintenance pricing: How much should you pay?* WebFX. Retrieved December 11, 2022, from https://www.webfx.com/web-development/pricing/website-maintenance/