|  |  |  |
| --- | --- | --- |
| **Project Name** Osric's Debuggers | | **Project Number** 0001 |
| **Project Start Date** 2018/05/23 | **Project End Date** 2018/08/20 | **Executive Sponsor** Thomas Steiner |
| * **Business Issue and / or Opportunity:** The Osric's company offers services in which there is a high demand for its technicians. Such high demand cannot be backed all the time given a restricted number of technicians available. Due to the long waiting time for customers, the Osric's system opportunity arises where the telecommunications company can better service their customers in a shorter period of time. * **Purpose:**    + **Correct Problem Description**: There is a need to ration the services offered by prioritizing the customers when they’re requesting a service. The service will provide data and statistics once its implementation is simulated. As a result, the data collected thorough the simulation of the service will be able to determine if to accept or reject Osric's scheme compared to a traditional “first-come-first-served” system.   + **Determine the efficiency of Osric's system by simulating it:**     - First, generate a job list (set of jobs). For each job, specify the necessary attributes (Time call is made, company priorities, and the duration) Other attributes are also included.     - When each job is called in, it's added to the queue.     - At the start of each block, jobs are removed from the queue and assigned to an available technician     - When a job is finished, the technician can go home or start another block. * **Current State:**   + **Priority System:**  1. Priority 4: Decorations for executive office 2. Priority 3: Company with 3+ previous calls 3. Priority 2: Company with 1-2 previous calls 4. Priority 1: First time callers 5. New customers are automatically assigned to priority 1 6. Waiting list is sorted by the priority system based on date and time (Queue) 7. The priority of a job is temporarily increased  * **Osric's Technicians:**  1. Each technician works 8 hours shifts 2. There are currently seven technicians during the day. At 8 A.M., the technicians are assigned the top 7 companies on the wait list. 3. Two technicians will be assigned during the night shift 4. Technicians are given a job card that had ID number and will be asked by the assistant upon job completion 5. Is reassigned a job at the end of every 4-hour block 6. When a technician completes their first hour block job, they can either start their second 4-hour block of the shift or go home.  * **Service:**  1. Technicians are available 24/7 2. No applicants will be accepted during night services 3. After confirming the job request, time estimation will be given by the assistant 4. Manager Approves:    1. Jobs.    2. Services.    3. New Company’s Requests. 5. Fees of the services will be charged accordingly depending on the provided shift    1. Day services will be charged for $480    2. Night services will be charged for $960  * **Assistant:**  1. Receive job approval from the manager for new job request 2. Contacting the customer 3. At 4 P.M. if a job is not completed, the assistant contacts the customer and asks them if they would like a night technician at double the rate. 4. If they would like the same technician to assist them in the morning. 5. Assign a priority to each job. 6. Update the following on the waiting list:    1. Available Jobs.    2. Priority.    3. Time and Dates.    4. Removes jobs from queue after being assigned to a technician. 7. Confirming the job. 8. Assign & report to the technician. 9. Supervise the night technicians 10. Mail the billing to the customer 11. Reassigns job for the technicians at noon and/or at the end of every 4-hour block  * **Customer:**  1. Contacts assistant. 2. Requests a job. 3. Is assigned by a technician. 4. Customer has:    1. Phone number.    2. Customer number.    3. Company name.    4. New customer information. 5. Customer Has the billing information.  * **Billing:**  1. Address 2. Amount due 3. Number of hours of services and technician shift 4. Priority Number 5. Company Name/Customer Name 6. Due dates.  * **Proposed Future State:** To ensure and achieve customers satisfaction of the services provided by OOA&D company, we must provide high quality work in less time. This has been fulfilled by OOA&D's remarkable ability to completely redecorate large executive suites in two days! Such achievement has gain OOA&D many customers recognition and much job requests than ever! However, due to the high number of job requests, OOA&D technicians are constantly in high demand, the wait time for a technician could sometimes take up to two days, such problem need to be address ASAP, if no action is taken this could potentially put the company credibility/reputation in permanent risk. The only solution to the problem is to create a priority number system, where customer is given/assigned with a priority number, when customers request for a service, with a provided priority number, the system will be able to decide where to put the customer in the waitlist for the next available technician. The priority system not only enhance the scheduling appointment functionality, it also shortens the wait time on requesting for technicians and enhance the billing interface. The team should be able to reach a definitive plan through communication and meetings, so that a foot print for a system simulating the requirements of Osric's can be implemented. Hence, a modelling and a testing are part of what the future has in hold for us as a group. Eventually, through the planning, modelling we will be able to construct the software and deploy it to the company. * **Project Goal (one paragraph summary of problem issue/solution)** The Osric’s company will show how effective the scheme is by using data and statistics on a job mix. It will identify and use the average waiting time before a job is started, the average queue length, the percentage of time the queue is empty on day and night, the number of blocks when a technician is idle, and the number of jobs that cannot be continued at night because no technician is available. Using this data on both schemes will make it clear which scheme is cost effective for Osric, which reduces the waiting time for the customer. | | |
| **Principal Project Objectives**   1. Software Project Plan. 2. Requirements Phase 3. Design. 4. Development (Processing) 5. Implementation and Source Code 6. Testing/Integration 7. Statistics Gathered 8. Conclusion 9. Plan for Change 10. Time & Budget Management | | |
| **Principal Project Deliverables** | | |
| **Objective 1 - Software Project Plan**         1. Propose a plan that aims to satisfy Osric's scheme         2. Describe the different states or scenarios of Osric's scheme         3. State the customer requirements that will attempt to satisfy the customer with the final product         4. Ensure that software reaches or exceeds Osric's expectations.         5. Describe the different objectives of the project, and the roles of the team members based on their strengths         6. Assign deadlines for different objectives         7. Asses risks, assumptions, and excludes from scope | | |
| **Objective 2 – Requirements Phase**          1. Find out the time, cost, resources and scope          2. Identify the requirements based on the assignment provided          3. Check with the customer for approval          4. Identify Osric's request          5. Identify the possible solutions          6. Researching and analyzing the different aspects of the project          7. Group sessions and strategic meetings          8. Setting up meetings with the customer to ensure quality          9. Brainstorm ideas on the possible requirements          10. Make sure the requirements are SMART – Specific, Measurable, Agreed upon, Realistic, and Time-Based          11. Avoid speaking about technology or solutions until the requirements are understood          12. Get the requirements agreed upon by the professor          13. Identify the different entities related the specific requirements          14. Discuss three additional functionalities to add to the current scheme          15. Gain clarity if there's any doubt          16. Avoid ambiguity | | |
| **Objective 3 – Design**   1. Complete ERD for the current state 2. Build ERD 3. Complete ERD after adding three new entities 4. Take into account time, resources, and cost to assemble information gathered during requirement gathering phase 5. Establish UML diagrams 6. Design mainframe 7. Choose the tools that will be used to develop the model 8. Come up with list that divides the primary tasks with the secondary tasks 9. Determine functions the software needs to perform 10. How functions will be implemented 11. Determine high level design of the system 12. Decide software and hardware to be used 13. Develop a program design (Breakdown the system into manageable modules) 14. Determine the different modules will interact with each other 15. Determine the system interface 16. Decide how the system will be tested 17. Estimate resources (develop a schedule and cost estimate) 18. Determine system documentation (user manual) 19. Plan for system installation 20. Implement using the Agile mythology 21. Research | | |
| **Objective 4 - Development (Processing)**   * 1. Start to develop the project once the design has been completed and approved   2. Implement the development cycle (process that describes approach to project/ 6 phase process)   3. Identify the different classes/entities and how they correlate with one another   4. Specify the different attributes for each entity   5. Identify the top priority tasks vs secondary tasks   6. Be able to come up with critical path   7. Assign each individual to a specific task along with a deadline of completion   8. Come up with a timeline to compete each development section   9. Ensure that every process that is intended to be produced is well documented by reviewing with one another   10. New risk assessments are made for the system   11. Develop a solution for each component   12. Set milestones for each development stage process   13. Project reviews sessions regarding the productivity and efficiency of the tasks   14. Enable customer to understand how the project will function and what it will look like   15. Refinement of the different designs established in the Design phase   16. Obtain approval to progress to testing phase from customer and executive sponsor | | |
| **Objective 5 - Implementation and Source Code**     * 1. The Implementation phase will begin once the Development phase has been approved and accepted   2. Build the solution   3. Apply a structured development approach   4. Ensure that the system is clear and understood by all of the coders   5. Develop a checklist with all the different components that need to be developed   6. Source code development   7. Identify the different skill set of each individual   8. Identify the language that the program will be coded in   9. Break the system into manageable pieces and in a simplified structure   10. Ensure that each programmer has a disciplined approach to the project   11. Identify possible re-usable code from past projects   12. Code the different classes, objects, and functions   13. Ensure that the different components work properly before going to the next task   14. Ensure that the basic functionalities are done before secondary one's   15. Keep up or update system changes   16. Ensure that the programmers do not lose sight of the big picture   17. Merge the all the different parts together   18. Ensure that each component recognizes and interacts with the system   19. Ensure that comments and descriptions are present throughout the source code   20. Ensure that the code has a clear formatting template   21. Debug software if necessary   22. Once the implementation and the source code has been completed, the next phase will begin with the testing and integration | | |
| **Objective 6 –Testing and Integration**   * 1. Approval to progress with project based on Implementation and source code completion   2. Check if the system is successfully built.   3. Test the development phase to see if it meets all requirements and design parameters   4. Proof read through the system and make sure the system meets all the requirements   5. The function regarding the design parameters must satisfy all business and technical needs   6. Assure that the system function is as described in the user manual or operations manual   7. Conduct system testing – test Analysis Report, Problem Report, and Test Analysis approval determination   8. System testing ensures to ensure that the system develops all technical and performance requirements   9. Static testing – conduct reviews, walkthroughs, and inspection   10. Keep track of any issues discovered   11. Test problem reports - documenting problems encountered during testing   12. Defect log - track the format defects or bugs   13. Dynamic testing – executing a set of code to test   14. Initiate testing activities – ensure that all data is loaded to test databases and prep any internal or external interfaces   15. Weekly tracking the progress of individual team member progress on assigned tasks   16. Test case development   17. Schedule of planned testing   18. Review of failed test cases   19. Make changes required to retest failed test cases   20. Create a Job Mix       1. There will be multiple versions of the job list.       2. Ensure that there will be multiple categories       3. For each job, there will be a different set of attributes tested       4. Time and date the call was made       5. The Priority of the company       6. Duration       7. Other attributes will be included as well   21. Readiness document – consolidates summary information based on the current status of the system and provides a decision makers information necessary to make a Go or No decision.   22. After testing is completed and accepted, the next phase will begin in the statistics gathering page | | |
| **Objective 7- Statistics Gathered**   1. Calculate the duration of each job 2. Queue based – First come first serve 3. Find the mean and standard deviation of the times taken for jobs to complete. 4. Mean and Standard deviation calculation for both systems 5. Highest and lowest times of competition 6. Time for each technician to complete a determinate job | | |
| **Objective 8 - Conclusion**   1. Determine Whether to Accept or Reject Osric's Scheme 2. Compare the data and evaluate if to reject, or if to accept 3. Accept: if the waiting time for the customers of Osric's is shorter than one-to-one & cost effective 4. Reject:  if waiting time is longer than the traditional first-come-first-serves scheme Present Demo to the customer | | |
| **Objective 9 - Plan for Change**   * + 1. Ensure that the team will be able to adapt to possible changes     2. Ensure that the team will be able to keep up with the different changes that will take place     3. Ensure that changes are made known to all team members     4. Ensure that changes are made known to the customer     5. Ensure that the team has planned for possible changes     6. Ensure that deadlines are re-edited and finalized once changes become effective     7. Ensure that changes made are effective for the system, and not burdensome | | |
| **Objective 10 - Time & Budget Management**   * **Time:**  1. The team should be able to complete the project within pre-established time constraints 2. The team should be able to plan every single step from beginning to end 3. The team should be able to plan for possible mistakes, and have time to adjust 4. The team should establish the most important tasks to be done first, and revolve secondary tasks around such tasks 5. The team should take a disciplined approach to the project by constantly checking the progress of the project and the timeline 6. The team should be able to build a timeline map in which all of the different tasks are outlined with respect to time 7. The team should be able to keep track of the actual time taken compared to the time planned for the various objectives  * **Budget:**   1. The team should be able to stay within a pre-determined budget   2. The team should be able to provide estimates to the customer on how much the project will cost   3. The team should be able to implement multiple techniques that will provide the necessary estimates   4. The team will have highly dependable individuals in key positions that deal with the management cost   5. The team should be able to pre-determine crucial and non-crucial expenses   6. The team should be able to track the actual cost vs estimated cost   **Quality:**   * 1. The project should be able to meet all the requirements   2. The project should be able to solve the problem   3. The project should be able to satisfy the customer's expectations | | |
| **Benefits (list hard and soft benefits)**   * More jobs completed in a day * Improved customer relationships * Increase of business * Faster response time for customers with higher priority * Tasks that are more important are completed earlier * Overall increase in revenue * Project team will gain experience | | |
| **Metrics to Measure Project Results**   * Number of jobs completed in a day, week, or month are based on a stack. * Monthly customers review of company's service | | |
| **Project Risks (details in risk table)**   * Ability to deliver desired scope within target. * Overall quality of data structure * Unsatisfied customers * Unplanned time constraints | | |
| **Related Projects**   * NA | | |
| **Project Teams/Support Teams**   * NA | | |

# 

# Project Scope

## Project Objectives & Deliverables Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective/ Deliverable Matrix** | **Responsible for Delivery** | **Supporting Team(s)** | **Receiving Team(s)** | **Delivery Date** |
| **Objective 1-** Software Project Plan |  |  |  | 6/4/2018 |
| **Deliverable 1.1:** Propose a plan that aims to satisfy Osric's scheme | Martin Zanaj | Team |  | 5/29/2018 |
| **Deliverable 1.2:** Describe the different states or scenarios of Osric's scheme | Maricruz Zamora | Team |  | 5/30/2018 |
| **Deliverable 1.3:** State the customer requirements that will attempt to satisfy the customer with the final product | Marco Seman | Team |  | 5/31/2018 |
| **Deliverable 1.4:** Ensure that software reaches or exceeds Osric's expectations. | Mohammed Almuzel | Team |  | 6/1/2018 |
| **Deliverable 1.5:** Describe the different objectives of the project, and the roles of the team members based on their strengths | Jason Lu | Team |  | 6/2/2018 |
| **Deliverable 1.6:** Assign deadlines for different objectives | Audrey White | Team |  | 6/3/2018 |
| **Deliverable 1.7:** Asses risks, assumptions, and excludes from scope | Marco Seman | Team |  | 6/4/2018 |
| **Objective 2** – Requirements Phase | Marco Seman | Team | TBA | TBA |
| **Deliverable 2.1:** Find out the time, cost, resources and scope |  |  |  |  |
| **Deliverable 2.2:** Identify the requirements based on the assignment provided |  |  |  |  |
| **Deliverable 2.3:** Check with the customer for approval |  |  |  |  |
| **Deliverable 2.4:** Identify Osric's request |  |  |  |  |
| **Deliverable 2.5:** Identify the possible solutions |  |  |  |  |
| **Deliverable 2.6:** Researching and analyzing the different aspects of the project |  |  |  |  |
| **Deliverable 2.7** Group sessions and strategic meetings |  |  |  |  |
| **Deliverable 2.8** Setting up meetings with the customer to ensure quality |  |  |  |  |
| **Deliverable 2.9** Brainstorm ideas on the possible  requirements |  |  |  |  |
| **Deliverable 2.10** Make sure the requirements are SMART – Specific, Measurable, Agreed upon, Realistic, and Time-Based |  |  |  |  |
| **Deliverable 2.11** void speaking about technology or solutions until the requirements are understood |  |  |  |  |
| **Deliverable 2.12** Get the requirements agreed upon by the professor |  |  |  |  |
| **Deliverable 2.13** Identify the different entities related the specific requirements |  |  |  |  |
| **Deliverable 2.14** Discuss three additional functionalities to add to the current scheme |  |  |  |  |
| **Deliverable 2.15** Gain clarity if there's any doubt |  |  |  |  |
| **Deliverable 2.16** Avoid  ambiguity |  |  |  |  |
| **Objective 3:** Design | Maricruz Zamora | Marco Seman | TBA | TBA |
| **Deliverable 3.1:** Complete ERD for the current state | Maricruz Zamora | Marco Seman | TBA | TBA |
| **Deliverable 3.2:** Build ERD | Maricruz Zamora | Martin Zanaj | TBA | TBA |
| **Deliverable 3.3:** Complete ERD after adding three new entities |  |  | TBA | TBA |
| **Deliverable 3.4:** Take into account time, resources, and cost to assemble information gathered during requirement gathering phase |  |  | TBA | TBA |
| **Deliverable 3.5:**  Establish UML diagrams | Maricruz Zamora |  | TBA | TBA |
| **Deliverable 3.6** Design mainframe |  |  |  |  |
| **Deliverable 3.7** Choose the tools that will be used to develop the model |  |  |  |  |
| **Deliverable 3.8** Come up with list that divides the primary tasks with the secondary tasks |  |  |  |  |
| **Deliverable 3.9** Determine functions the software needs to perform |  |  |  |  |
| **Deliverable 3.10** How functions will be implemented |  |  |  |  |
| **Deliverable 3.11** Determine high level design of the system |  |  |  |  |
| **Deliverable 3.12** Decide software and hardware to be used |  |  |  |  |
| **Deliverable 3.13** Develop a program design (Breakdown the system into manageable modules) |  |  |  |  |
| **Deliverable 3.14** Determine the different modules will interact with each other |  |  |  |  |
| **Deliverable 3.15** Determine the system interface |  |  |  |  |
| **Deliverable 3.16** Decide how the system will be tested |  |  |  |  |
| **Deliverable 3.17** Estimate resources (develop a schedule and cost estimate) |  |  |  |  |
| **Deliverable 3.18** Determine system documentation (user manual) |  |  |  |  |
| **Deliverable 3.19** Plan for system installation |  |  |  |  |
| **Deliverable 3.20** Implement using the Agile mythology |  |  |  |  |
| **Deliverable 3.21** Research |  |  | TBA | TBA |
| **Objective 4:** Development (Processing) | Jason Lu | Mohammed | TBA | TBA |
| **Deliverable 4.1:** Start to develop the design has been completed and approved | Jason Lu | Mohammed | TBA | TBA |
| **Deliverable 4.2:** Implement the development cycle (process that describes approach to project/ 6 phase process) | Maricruz Zamora |  | TBA | TBA |
| **Deliverable 4.3:** Identify the different classes/entities and how they correlate with one another |  |  |  |  |
| **Deliverable 4.4:** Specify the different attributes for each entity |  |  |  |  |
| **Deliverable 4.5:** Identify the top priority tasks vs secondary tasks |  |  |  |  |
| **Deliverable 4.6:** Be able to come up with critical path |  |  |  |  |
| **Deliverable 4.7:** Assign each individual to a specific task  along with a deadline of  completion |  |  |  |  |
| **Deliverable 4.8:** Come up with a timeline to compete each  development section |  |  |  |  |
| **Deliverable 4.9:** Ensure that every process that is intended to be produced is well documented by reviewing with one another |  |  |  |  |
| **Deliverable 4.10:** New risk  assessments are made for the  system |  |  |  |  |
| **Deliverable 4.11:** Develop a  solution for each component |  |  |  |  |
| **Deliverable 4.12:** Set milestones for each development stage process |  |  |  |  |
| **Deliverable 4.13:** Project reviews sessions regarding the productivity and efficiency of the tasks |  |  |  |  |
| **Deliverable 4.14:** Enable customer to understand how the project will function and what it will look like |  |  |  |  |
| **Deliverable 4.15:** Refinement of the different designs established in the Design phase |  |  |  |  |
| **Deliverable 4.16:** Obtain approval to progress to testing phase  From customer and executive  sponsor |  |  |  |  |
| **Objective 5:** Implementation  and Source Code | Mohammed | Marco Seman | TBA | TBA |
| **Deliverable 5.1:** The  implementation phase will begin once the Development phase has been approved and accepted |  |  |  |  |
| **Deliverable 5.2:** Build the solution |  |  |  |  |
| **Deliverable 5.3:** Apply a  structured development  approach |  |  |  |  |
| **Deliverable 5.4:** Ensure that the system is clear and  understood by all of the coders |  |  |  |  |
| **Deliverable 5.5:** Develop a checklist with all the different components that need to be developed |  |  |  |  |
| **Deliverable 5.6:** Source code  development | Mohammed | Marco Seman | TBA | TBA |
| **Deliverable 5.7:** Identify the different skill set of each individual |  |  |  |  |
| **Deliverable 5.8:** Identify the language that the program will be coded in |  |  |  |  |
| **Deliverable 5.9:** Break the system into manageable pieces and in a simplified  structure |  |  |  |  |
| **Deliverable 5.10:** Ensure that each programmer has a disciplined approach to the project |  |  |  |  |
| **Deliverable 5.11:** Identify possible reusable code from past projects |  |  |  |  |
| **Deliverable 5.12:** Code the  different classes, objects, and functions |  |  |  |  |
| **Deliverable 5.13:** Ensure that  the different components work  properly before going to the next task |  |  |  |  |
| **Deliverable 5.14:** Ensure that  the basic functionalities are done before secondary one's |  |  |  |  |
| **Deliverable 5.15:** Keep up or update system changes |  |  |  |  |
| **Deliverable 5.16:** Ensure that the programmers do not lose  sight of the big picture |  |  |  |  |
| **Deliverable 5.17:** Merge the  all the different parts together |  |  |  |  |
| **Deliverable 5.18:** Ensure that each component recognizes and interacts with the system |  |  |  |  |
| **Deliverable 5.19:** Ensure that comments and descriptions are present throughout the  source code |  |  |  |  |
| **Deliverable 5.20:** Ensure that the code has a clear formatting  template |  |  |  |  |
| **Deliverable 5.21:** Debug software if necessary |  |  |  |  |
| **Deliverable 5.22:** Once the implementation and the source  code has been completed, the next phase will begin with the  testing and integration |  |  |  |  |
| **Objective 6:** Testing and Integration | Marco Seman | Martin Zanaj | TBA | TBA |
| **Deliverable 6.1:** Approval to progress with project based on Implementation and source code completion |  |  |  |  |
| **Deliverable 6.2:** Check if the system is  successfully built. |  |  |  |  |
| **Deliverable 6.3:** Test the development phase to see if it meets all requirements and design parameters |  |  |  |  |
| **Deliverable 6.4:** Proof read through the system  and make sure the system meets all the requirements |  |  |  |  |
| **Deliverable 6.5:** The function regarding the design parameters must satisfy all  business and technical needs |  |  |  |  |
| **Deliverable 6.6:** Assure that the system function is as  described in the user manual or operations manual |  |  |  |  |
| **Deliverable 6.7:** Conduct system testing –test Analysis Report, Problem Report, and Test Analysis approval determination |  |  |  |  |
| **Deliverable 6.8:** System testing ensures to ensure that the system develops all technical and performance requirements |  |  |  |  |
| **Deliverable 6.9:** Static testing – conduct reviews, walkthroughs, and inspection |  |  |  |  |
| **Deliverable 6.10:** Keep track of any issues discovered |  |  |  |  |
| **Deliverable 6.11:** Test  problem reports - documenting  problems encountered during  testing |  |  |  |  |
| **Deliverable 6.12:** Defect log - track the format defects or bugs |  |  |  |  |
| **Deliverable 6.13:** Dynamic testing – executing a set of code to test |  |  |  |  |
| **Deliverable 6.14:** Initiate testing activities – ensure that all data is loaded to test  databases and prep any internal or external interfaces |  |  |  |  |
| **Deliverable 6.15:** Weekly tracking the progress of individual team member progress on assigned tasks |  |  |  |  |
| **Deliverable 6.16:** Test case development |  |  |  |  |
| **Deliverable 6.17:** Schedule of planned testing |  |  |  |  |
| **Deliverable 6.18:** Review of failed test cases |  |  |  |  |
| **Deliverable 6.19:** Make changes required to retest failed test cases |  |  |  |  |
| **Deliverable 6.20:** Create a Job Mix |  |  |  |  |
| **Deliverable 6.20.1:** There will be multiple versions of the job list. |  |  |  |  |
| **Deliverable 6.20.2:** Ensure that there will be multiple categories |  |  |  |  |
| **Deliverable 6.20.3:** For each job, there will be a different set of attributes tested |  |  |  |  |
| **Deliverable 6.20.4:** Time and date the call was made |  |  |  |  |
| **Deliverable 6.20.5:** The Priority of the company |  |  |  |  |
| **Deliverable 6.20.6:** Duration |  |  |  |  |
| **Deliverable 6.20.7:** Other attributes will be included as well |  |  |  |  |
| **Deliverable 6.21:** Readiness document- consolidates summary information based on the  current status of the system and provides a decision makers information necessary to make a Go or No decision. |  |  |  |  |
| **Deliverable 6.22:** After testing is completed and accepted, the next phase will begin in the  statistics gathering page |  |  |  |  |
| **Objective 7:** Statistics  Gathered | Marco Seman | Audrey White | TBA | TBA |
| **Deliverable 7.1:** Calculate the duration of each job |  |  |  |  |
| **Deliverable 7.2:** Queue based – First come first serve |  |  |  |  |
| **Deliverable 7.3:** Find the  mean and standard deviation of the times taken for jobs to complete. |  |  |  |  |
| **Deliverable 7.4:** Mean and  Standard deviation calculation for both systems |  |  |  |  |
| **Deliverable 7.5:** Highest and lowest times of competition |  |  |  |  |
| **Deliverable 7.6:** Time for each technician to complete a determinate job |  |  |  |  |
| **Objective 8:** Conclusion | Audrey White | Martin Zanaj | TBA | TBA |
| **Deliverable 8.1:** Determine Whether to Accept or Reject  Osric's Scheme |  |  |  |  |
| **Deliverable 8.2:** Compare the data and evaluate if to reject, or if to accept |  |  |  |  |
| **Deliverable 8.3:** Accept: if the waiting time for the customers of Osric's is shorter than one-to-one & cost effective |  |  |  |  |
| **Deliverable 8.4:** Reject: if  waiting time is longer than the traditional first-come-first-serves scheme |  |  |  |  |
| **Deliverable 8.5:** Present Demo to the customer |  |  |  |  |
| **Object 9:** Plan for Change | Maricruz Zamora |  | TBA | TBA |
| **Deliverable 9.1:** Ensure that the team will be able to adapt to possible changes |  |  |  |  |
| **Deliverable 9.2:** Ensure that the team will be able to keep up with the different changes that will take place |  |  |  |  |
| **Deliverable 9.3:**  Ensure that changes are made known to all team members |  |  |  |  |
| **Deliverable 9.4:** Ensure that changes are made known to the customer |  |  |  |  |
| **Deliverable 9.5:**  Ensure that the team has planned for possible changes |  |  |  |  |
| **Deliverable 9.6:** Ensure that deadlines are  Re-edited and finalized once  changes become effective |  |  |  |  |
| **Deliverable 9.7:**  Ensure that changes made are effective for the system, and not  burdensome |  |  |  |  |
| **Objective 10:** Time & Budget Management | Maricruz Zamora | Marco Seman | 07/01/2018 | 08/20/2018 |
| **Time Deliverable 10.1:** The team should be able to complete the project within pre-established time constraints |  |  |  |  |
| **Time Deliverable 10.2:** The team should be able to plan every single step from beginning to end |  |  |  |  |
| **Time Deliverable 10.3:** The team should be able to plan for possible mistakes, and have time to adjust |  |  |  |  |
| **Time Deliverable 10.4:** The team should establish the most important tasks to be done first, and revolve secondary tasks around such tasks |  |  |  |  |
| **Time Deliverable 10.5:** The team should take a disciplined approach to the project by constantly checking the progress of the project and the timeline |  |  |  |  |
| **Time Deliverable 10.6:** The team should be able to build a timeline map in which all of the different tasks are outlined with respect to time |  |  |  |  |
| **Time Deliverable 10.7:** The team should be able to keep track of the actual time taken compared to the time planned for the various objectives |  |  |  |  |
| **Budget Deliverable 10.1:** The team should be able to stay within a pre-determined budget |  |  |  |  |
| **Budget Deliverable 10.2:** The team should be able to provide estimates to the customer on how much the project will cost |  |  |  |  |
| **Budget Deliverable 10.3:** The team should be able to implement multiple techniques that will provide the necessary estimates |  |  |  |  |
| **Budget Deliverable 10.4:** The team will have highly dependable individuals in key positions that deal with the management cost |  |  |  |  |
| **Budget Deliverable 10.5:** The team should be able to pre-determine crucial and non-crucial expenses |  |  |  |  |
| **Budget Deliverable 10.6:** The team should be able to track the actual cost vs estimated cost | Maricruz Zamora | Marco Seman |  | 08/20/2018 |
| **Budget Deliverable 10.7:** The project should be able to meet all the requirements |  |  |  |  |
| **Budget Deliverable 10.8:** The project should be able to solve the problem |  |  |  |  |
| **Budget Deliverable 10.9:** The project should be able to satisfy the customer's expectations |  |  |  |  |

## Exclude from Project

* We are not adding new functionalities
* New fully integrated software.
* Website design

## Assumptions

* We will have access to customers' information and requests.
* Availability, completeness, and quality information are the data we will be working with (ex. Customer name, customer number, orders, etc.)

## External Time Constraints

* Not receiving data from depending team
* Resource shortage
* System upgrades
* The end of the semester
* Deadline for specific tasks

## Financials (Costs and Benefits)

See Business Case.

(This will be determined later)

**Risk Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability of Risk H/M/L\*** | **Potential Impact of Risk C/H/M/L\*** | **Risk Mitigation Strategy**  **and Contingent Action(s)** |
| Team member unavailability (Include conflict scheduling, summer jobs, etc.) | H | M | Mitigation: Meet with those who are available, then inform those who weren't able to meet. Assign multiple group members to required tasks. Also, we could do video calling to discuss and finish the tasks.  Contingency: find a replacement as soon as possible. If not one single individual will be able to achieve the task at hand, the tasks will be completed by all the different members in a way such that the amount of work is equally divided. |
| Software and Hardware problems | M | L | Mitigation: Such problem can be avoided by using reliable services and by saving our work on multiple drives.  Contingency: Use the resources that are provided at school |
| Sickness | M | L | Mitigation: Ensure that each individual is not living a reckless life (at least until the course is over)  Contingency: There will be a partner associated with each member, and if one of the two individuals comes to miss the partner can update the others on the current situation and seek help if needed. |
| Miscommunication | H | M | Mitigation: If team members are unable to resolve grievances via GroupMe, they should address any concerns they may have during weekly meetings.  Contingency: If miscommunications on a specific part of the project continue regularly, the team lead and rest of team should reassess the issue and find a resolution all members agree on. |
| Changes in project scope | H | C | Mitigation: If changes need to be made in the project, the team member should first report the changes to the team lead first and should not make any changes without reporting.  Contingency: If the changes cannot be avoided, problems with project shift changes will be resolved as a team. |
| Missed deadlines and lack of time | M | H | Mitigation: We will setup milestones in advance of the final due date for each deliverable in order to be more organized and making sure to finish on time. Team members should plan their schedule accordingly in order to complete their assignments by the deadline and not take on more work than they can handle during that time frame.  Contingency: If someone is unable to complete their work for that week, then others with less work assigned may take on some of their assignments. If a team member continues to miss deadlines, there will be a team discussion regarding a time management plan for the team member. |
| Incorrect software design | L | H | Mitigation: Design documents for each aspect of the software should be approved by team member responsible and team lead.  Contingency: Errors in design documentation will be reviewed and any changes made will be revised prior to software implementation to avoid design errors in final product. |
| Insufficient experience with current technologies (Type of tool and software) | M | M | Mitigation: Since the level of experience varies between team members, it is important that less experienced team members ask questions and are honest when they do not understand relevant technology, software, etc.  Contingency: If someone is unqualified for an assignment due to lack of technical experience, the assignment will also be assigned at least one member with prior experience. |
| Undiscovered defect | M | M | Mitigation: To minimize the number of errors that can be caused by the user input, team Debugger will review every possible input that the user could enter.  Contingency: During the development phase, every test will be documented. If a new defect is found during the development phase, the defect will also be documented. |
| Changing in the requirements | M | C | Mitigation: There may be some changes that will come, so we need to assign extra time before the deadline to deal with this kind of issues in advance.  Contingency: If changes do occur, we will priotize what can be shifted on the project time line to make sure the project stays within proposed time and budget. |
| All know different programming language | M | M | Mitigation: We must use programming languages that many of us are familiar with.  Contingency: If someone is unqualified for a programming or testing assignment due to their lack of knowledge of a particular language, the assignment will also be assigned to at least one member fluent in that language. |

\* Critical /High / Medium / Low

## Reporting – Scope Change, Issue, Risk Management (RMMM) and Status

* **Scope Change:**
  + There have been no changes to the features as of May 28, 2018 (Making adjustments to the cost, budgets, other features, or the timeline)
  + On May 30th, the objectives and deliverables were updated.
  + On June 3rd and 4th, we revamped our objectives and project scope
  + On June 4th we updated our risk analysis to make it more clear
* **Issues:**
  + User Interface idea
  + Connecting all objectives together
  + Knowledge on programming languages and user interface design
  + Constantly change of requirements
* **Risk Managements:**
  + **Risk:** 
    - Client doesn't like what was created
    - Client could be slow with communication
    - Client changing their request
    - Meeting deadlines
    - Managing and collaborating with team members
* **Monitoring: project risks open** 
  + **Management:** 
    - If there is any member of our group that is unable to meet during a scheduled meeting time, then that person is responsible for tasks that missed and will receive negative feedback from the team leader and from other members if no valid excuse is made.
    - The team leader for our group will be responsible for contacting client frequently and informing them of any updates; at least once a week, when a major task is completed, or if a problem occurs
* **Status:**
  + An update of the project's status will be reported weekly to the Executive Sponsor.
  + The team will also have a weekly meeting to review the status and progress of the project.

# Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Information Type** | **Subject** | **Responsible Person** | **Frequency** | **Audience** |
| Report | Weekly Status | Team Lead | Twice a week (Monday's and Wednesday's) | Project Manager |
| Core team meeting through Skype | Discuss project goals | Project Manager | Saturday | Core team members |
| Emails | ideas/questions | Team members | Any | Team Members |
| GroupMe | Discuss project related matters | Team members | Any | Team Members |
| OneDrive Report on Project Deliverables | Report Approval | All | Any | All |

# Project Organization

## Roles and Responsibilities - Project Team Members & Support Personnel

|  |  |  |
| --- | --- | --- |
| **Project Role** | **Name or skill set required** | **Department/Organization** |
| Project Manager | Project management | Project Management |
| System Analyst | Business/IT Application knowledge | Application IT team |
| Business Team | Business Analyst | Business Department |
| Web design Team | Web development | Web design department |
| Software development | Programming language | Software development department |
| Marketing | Market Research | Marketing department |
| Sales | Customer Target | Sales department |
| Technology | Resources | Tech department |
| Training | Specific Skills | Human Resources |
| Technical Writing | Technical Writing | Customer Servies Department |
| IT department | Unit/Integration Testing | IT department |
| Software analyst | Some type of programming knowledge | ID department |
| Risk management | Risk management | Risk Management |

# Project Approval Signatures

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Project Team Role** | **Signature** | **Date** |
| Marco Seman | Team Leader | 𝑀𝒶𝓇𝒸𝑜 𝒮𝑒𝓂𝒶𝓃 | 05/30/2018 |
| Martin Zanaj | Project Manager | 𝑀𝒶𝓇𝓉𝒾𝓃 𝒵𝒶𝓃𝒶𝒿 | 05/30/2018 |
| Mohammed AlMuzel | Software Development |  | 2018/06/04 |
| Audrey White | IT Department, Testing | 𝒜𝓊𝒹𝓇𝑒𝓎 𝒲𝒽𝒾𝓉𝑒 | 05/30/2018 |
| Maricruz Zamora | System Analyst | 𝑀𝒶𝓇𝒾𝒸𝓇𝓊𝓏 𝒵𝒶𝓂𝑜𝓇𝒶 | 05/30/2018 |
| Jason Lu | Data Structures | 𝒥𝒶𝓈𝑜𝓃 𝐿𝓊 | 05/30/2018 |

# Document Information

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Revision Notes** |
| 1.0 | May 22, 2018 | Jason Lu, Marco Seman, Mohammed AlMuzel, Martin, Maricruz Zamora, Audrey White | This is the start of our documentation |
| 1.1 | May 29, 2018 | Jason Lu, Marco Seman, Mohammed AlMuzel, Martin, Maricruz Zamora, Audrey White | Further Develop our rough draft. Updated the objectives and project scope |
| 1.2 | June 3, 2018 | Team | Revamped our objectives, project scope, and risk analysis |