08

**Fall**

Prof: Tamara Blake

Project by:

Veena Chintala

And

Sreekanth Arun Daripally

Golden Gate University

ITM 316

Final Project

Fall 2018

Admissions Data Processing System

**TABLE OF CONTENT**

**1. Introduction……………………………………………………………………………… 1**

**2. Project Management...……………………………………..……………………………..1**

**3. Introduction to Scrum team….……………………………………………......................2**

**4. Project Plan …………………………………………….…………………………........... 4**

**5. Agile Team**:**………….…………………………….………………………………............4**

**6. Role and Responsibilities**: **………………………………………………………….……..5**

**7. Team dynamics:…………………………………………………………..………………. 5**

**8. Software Requirements…………………………………………………………………. 6**

**9. Non - Functional requirements…………..…………………………………………….. 10**

**10. Functional requirements……………….……………………………………………… 11**

**11. System design and Architecture.………………………………………………………14**

**12. Change Management………………………………………………………………….. 14**

**13. Unit Testing ………..………………………………………………………………….. 15**

**14. Release testing ……..………………………………………………………………….. 16**

**15. Unit testing ………....………………………………………………………………….. 16**

**16. Project Plan - Timelines……………………………………………………………….. 20**

**17. Deployment…………….……………………………………………………………….. 21**

**18. Version Controlling ….………………………………………………………………..  22**

**19. Future Plan ….………………...……………………………………………………….. 23**

**20. Reference …....………………...……………………………………………………….. 24**

**Introduction**

Organizations involved in various businesses across the globe functions based on their potential customers and their product development plans. Driving product development and services require metrics i.e. numbers. A product based company focuses on manufactured volume vs demand in the market. Therefore numbers in terms of production and sales are captured in a timely manner. These numbers play a vital role in making business decisions to identify areas that need attention or improvements. Irrespective of the type of business, metrics help business to identify the influencing factors of customer behavior and also to predict volumes for the upcoming quarter or yearly volumes. It is easier for a human brain to process information quickly using graphs and visual charts. Visualizing the complex numbers using graphs or charts makes it easier than presenting them in spreadsheets. Data visualization is quicker with the help of modern techniques.

Making the right decisions at the right time will enable the business to compete within the industry. One of the best ways to do it is by analyzing the data of the customers. According to Young (2017), Business Analytics tools, dashboards visualizing data are very helpful in analyzing the data and providing valuable business insights. Using this analysis the top management can make a competitive and successful decision which can help improve the business.

Educational institutions like Golden Gate University are well known for the students enrolling from various parts of the world for various programs. The current and historical admissions data is huge for all of the courses offered at GGU. It is been identified and taken an opportunity to provide a dashboard for visualizing the previous and current admissions data based on yearly and quarterly terms verses programs offered at the university. With the new Admission Data Processing (ADP) system, it easier to compare, visualize and assess the students (Internation, Domestic and Veteran) enrolling in various programs offered at GGU.

GGU Admin department will use our tool to analyse the behaviour of students joining the college. The tool helps to view the number of students admitted and enrolled between a time period. The tool also provides a forecast the enrollment of students for next 5 years.

Mission statement:

Solve the unsolved with the power of data and analytics.

**Project Management**

Project management is the practice of planning, initiating, controlling, executing, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time. This project requires to gather and understand requirements before starting the development process, hence we are using combination of Waterfall and Agile methodology.

Agile method is a incremental development method where small increments are developed and released in every 2 or 3 weeks. We are planning to use Scrum as our project management framework with a 2 week sprint.

**Introduction to Scrum team**

We are a team of 6 people, with one product owner, one scrum master, two developers and a tester. The requirements are broken down into smaller tasks and added into backlog by the product owner during the initial stage of the project. We start our sprint with a sprint planning meeting, where product owner presents the list of tasks to be worked on for that sprint. The team picks the tasks for the current sprint which they think they can complete. Now the tasks will be moved from product backlog to sprint backlog. During the sprint we have daily stand up where we discuss on what we are accomplishing for that day and if there are any roadblocks. On the last day of sprint we will have Sprint review meeting to demonstrate what the team has achieved during the sprint. Lastly, we will have sprint retrospective where we discuss on issues or changes the team wish to make. All the request, during the sprint, for the feature change or updates, if not of high priority are taken in the next sprint. Each feature is documented in a shared folder which is accessible by everyone so that it is transparent to the team and easier to refer, if necessary in the long run.

We will be including Test driven development where we write the test cases before writing the code. The test development requires a clear understanding of the requirements as the test cases written should match the system requirements that are implemented using code. The system requirements are broken down into smaller tasks on which the test cases are written.

We will be using Sql as backend and AWS for maintaining services. Code development is done using jquery and python languages. Microstrategy is used for Visualization. Additionally, we will be testing for the performance and security of the system.

ADC Product Backlog

ADC Sprint Backlog

ADC Software ready to release

Select Items

Sprint

Scrum

Review Sprint

Review work to be done

**Waterfall model for Requirements engineering and Approval process**

As there is no one precise model for project management, the Waterfall model can be used during our initial phase of requirements engineering and approvals process. ADP system's main goal is to present that admissions data in a visualized fashion. The main core requirements are admissions data ( approximately for last 5 years) and Dashboard for users. Users of the ADP are identified in 3 levels. Admissions processing team who directly work with students, Admissions data analyst, and the top management to make business decisions.

Each level os user of ADP needs a different level of access to the system.

*Waterfall Model - Requirements engineering and Approval*

Admission Data Processing System Proposal

Submit Design  
(Technical/Functional)

Quit Project

Project Approval

NO

YES

**Project Plan**

Introduction

Our initial phase of requirements gathering and project approvals must be well documented along with the goal of the project. The main objective of the project is to have a visualized data representation for ease of making business decisions. With the new Admissions data Processing (ADP) system, users will have the feasibility to access the visualized data in terms of bar graphs and pie charts. Graphs representing yesteryears' graphs and current student enrollment. Entire project requires to have a Data warehouse setup and a dashboard system. The budget and time are the main constraints of the project. Due to the complexity of the backend infrastructure, the entire team should follow the agile methodology to accomplish the project.

**Agile Team**:

Extreme programming (XP) would be the right choice to drive the project within timelines. A highly skilled team is required to accomplish the project. With the use of extreme programming risk mitigation is easier at any point in time (Sergeev, 2018). Each team member should be well versed with programming, databases, application servers, operating systems and networking skills. Each team member will be given equal opportunity in each area while implementing the project.

Even though the complexity of the project requires a large team, this project can also be achievable with a highly skilled small team.

***Role and Responsibilities***:

***Scrum Master***: Along with technical contribution as team lead, Scrum master in Scrum act as a project lead in various other methods, This role is also responsible for engaging the and facilitating the team by obtaining required resources during the development process. This role has the skills of project management but not the technical ones such as planning and scheduling the tasks.

***Team member***: As a team member also referred to as a developer is responsible for the creation by coding and delivery of a system. Team member roles’ main job is programming, testing, and release activities, as well as support activities.

***Product owner:*** In extreme programming model (XP), the Product owner will be one of the employees from the Admissions department. The employee from the Admissions department will be closely working with the agile team. From product initiation i.e. requirements gathering to the deployment phase, Product owner will be involved in prioritizing the work i.e. product backlog and he/she is responsible for providing inputs based on the changing business needs. It is also the Product owner's responsibility to assess the overall functionality of the system in a timely manner.

***Stakeholder***: Golden Gate University’s employee working in Admission processing team, Higher manager management whose decisions are impacting the business and one who funding the pare the main stakeholders of this project.

***Workplace***:

Product owner needs higher visibility of progress of the project. It is decided to have the team on campus to for the entire project time frame. The team will be functioning from morning 9 AM to evening 5.30PM during the weekdays. Product owners are stakeholders have the feasibility to communicate with the team for the modifications and enhancements. Since the time, scope and budget are expected to be limited, having virtual team may hinder the delivery process. And other main reason is that this project comprises of a small team six members only.

***Team dynamics:***

To have fast pace delivery and quality outcome, We would strive to have open and honest communications, Our activities will build trustworthiness among team members which helps us as a team to become a functional team. To effectively manage a team by standardizing and aligning the team member short-term goals with the project's vision and mission requires strong leadership as well as managerial skills. Following are the most important aspects, which needs to implement carefully to manage this agile team

*Trustworthiness*:

When a group of individuals working as a team, it is obvious that misunderstanding arises in various situations. People with issues, concerns are common within a team. They may not share information with one another. It doesn't matter how capable and talented people are, still without trust they may not perform well. We need to show the team members that you trust others. Setting examples by trusting your team, colleagues, and your vertical head is the only way. Having regular one-on-one meetings, understanding the problems and achievements and appreciating their goals play an important role while building trust.

*Honest and Timely Communication*:

Clear and honest communication is another important aspect of managing a team. Creating an environment for team members to talk honestly and in a meaningful way is essential to building trust within team members. Timely communications about the team progress, where the team is going, how each individual is performing are important. Emphasizing company's vision and the team's contribution makes the team members understand their importance. Constantly providing feedback on behavior individually, making them understand expectations and guiding each individual whenever required creates a joyful atmosphere.

*Quality delivery*:

Everybody makes mistakes. But learning lessons from mistakes and not doing them again is wise to manage the failures. When team members make mistakes, instead of punishing, we need to explain them the negative impact. Encourage team members to do mistakes. Offering new opportunities, training them on new skillset and creating an environment for participation motivate the team members to remain competent. Career development path gives develops trust among the team members.

**Software Requirements**

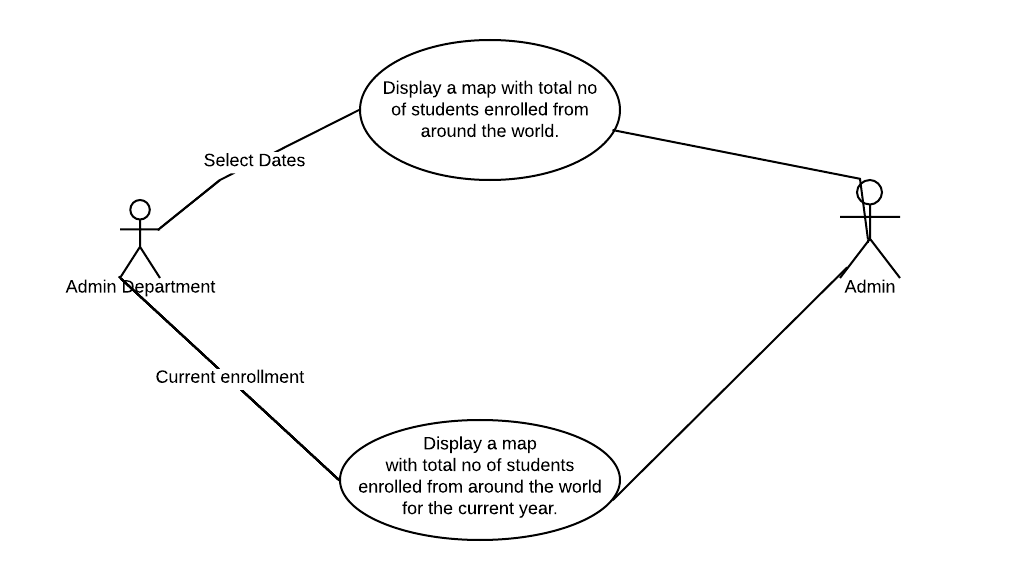
In our project we are using Sql server for backend and Python for writing code. Microstrategy tool is used for visualization. Github is used to store the application at a common location. Jenkins is used to merge the code and all this is in AWS cloud.

**Non - Functional requirements**

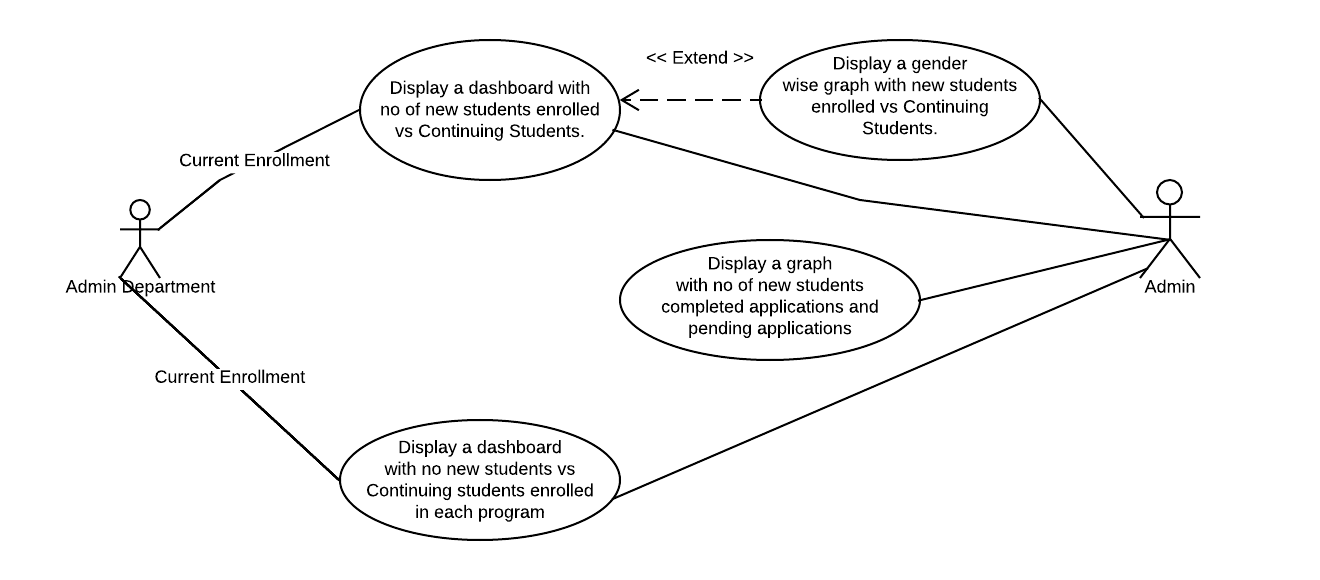
In our application decision maker can make a comment based on changes in law, government or university changes. If the data is not periodically updated by the admissions team, the dashboards would display incorrect results. User need Internet to access the application. Also he needs to be authorized to access the application. For the continuous availability of the system the application is deployed in two servers.

**Functional requirement use cases**

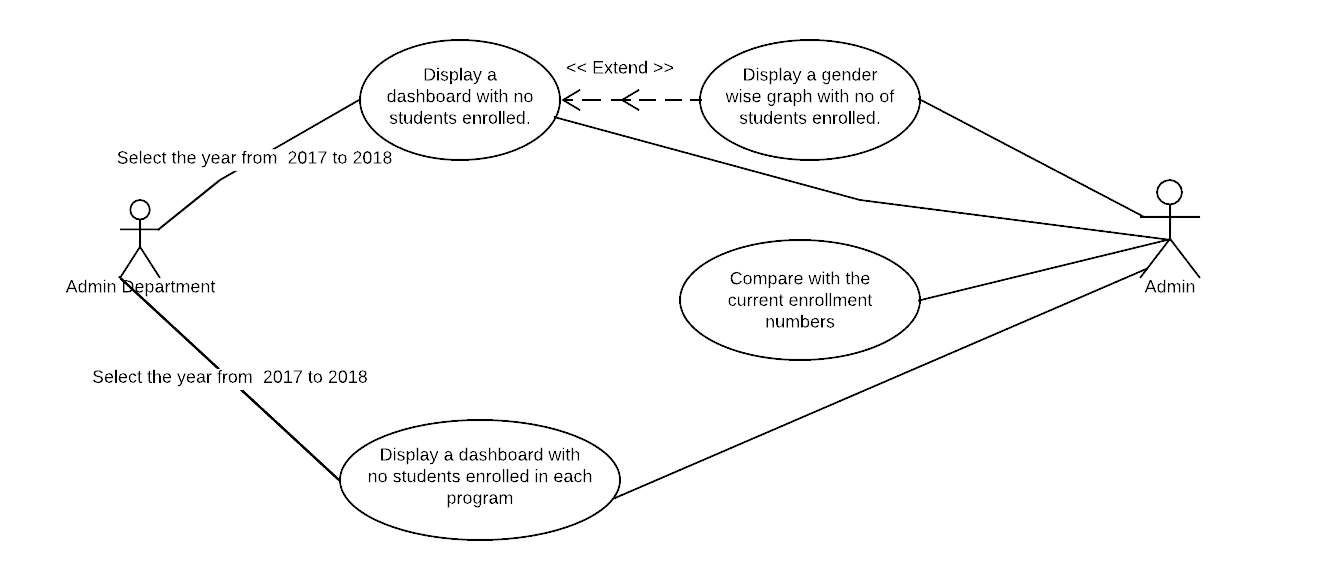
**Use cases For Administration Department**



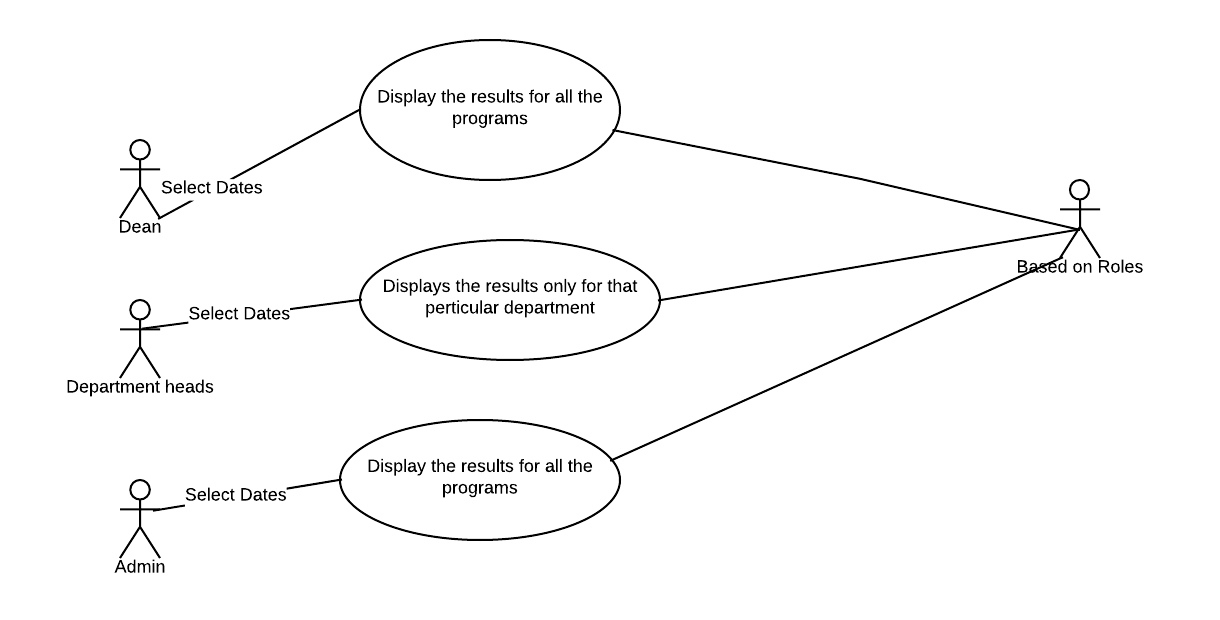
**Accessing application by selecting current year data**



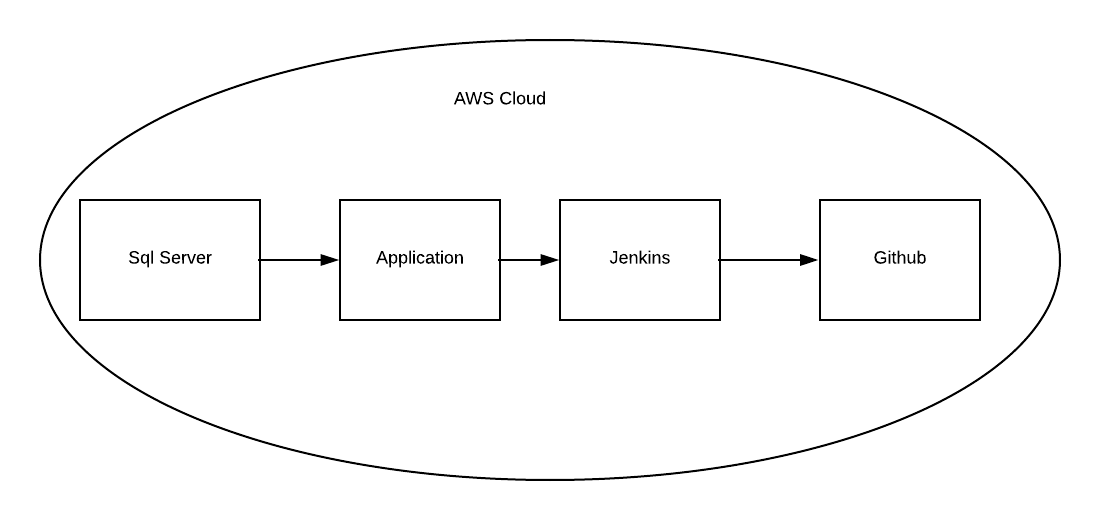
**Accessing application by selecting previous year data**

****

**Application accessibility use case based on roles**



**System design and Architecture:**



**Change Management**

Changes are inevitable in a project. The changes may occur due to legislative change(federal or State law), business change or customer changes. Product owner may have realized what the customer had asked for and would request the team for the respective changes. Agile development process accommodates and encourages the changes in the application as a project develops. Therefore in Agile the design, development, planning and testing keeps changing as the project continues to proceed.

In our project we take up the changes with a certain rules. After the sprint planning during the sprint any request for changes is not accepted. All the changes will be put in the backlog and taken up according to the priority from the backlog list in the next sprint. All the team members including customers and stakeholders are involved in the Sprint meetings at the end of the sprint. During the demo session customers are encouraged to provide their feedback and propose any changes they need in the project. But, any priority issues related to security will be taken up with an immediate action in the sprint.

Any change requests will be documented and shared in a common place where it is accessible by everyone. To avoid spending lot of time on documentation, if the team receives an email for the change request the mail content itself will be saved in the shared folder for the record.

ADC Product Backlog

ADC Sprint Backlog

ADC Software ready to release

Select Items

Sprint

Scrum

Review Sprint

Review work to be done

Customer Change Request (CR)

CR Approved by Product Owner

**Unit Testing**

Testing is a very important process and necessary to be conducted before releasing the project to the users to check if all the functionalities are working as expected.

According to Sherman (2018), Agile testing is a software testing process that is inclined with the principles of agile software development. Since Agile is an incremental process, testing is included in every sprint. In Agile, testing is not only done by testers but also by developers and business analysts. All the defects raised by the team is fixed in the same sprint to keep the code clean.

We are planning to use all the testing process for our project.

1. Development testing: Unit test cases are written to test the functionality of the methods and objects. Test cases are also written to test various components in the project. Development test is used to find the bugs in the application, it is nothing but the defect testing process.

2. Test- driven development: Test cases are written even before writing the code for the feature. There are two test cases written for every feature one to fail and the other to pass. This can help in refactoring the code. Testing should be repeated whenever there is a change in code. Test driven development drives repetitive testing. This repetitive testing can ensure that code is working perfectly in every step.

Add a ADP developer test case

Run the developer ADP tests

Change the developer test case

Re-run the ADP tests

[Pass]

[Fail]

[Fail]

[Pass, Development continues]

[Pass, Development stops]

**3. Release testing :**

Release testing which is also called as functionality testing is used to test the functionality of the system. This testing do not concerned about implementation it only concerned about the functionality of the application. In our project release testing is used to test the functionality of the ADP system.

**Release testing - plan**

The team will be performing the release planning during the sprint zero, where product increments are no more delivered. this sprint is completely dedicated for planning next releases. In this sprint we will be working on deciding what features need to be released for the customers, and defining release backlogs. Features are breaked into small user stories and acceptance criteria is written for these stories. We will be modifying the release plan based the user stories added or deleted over time.

**4. User testing:**

User testing is the process where customer provides certain inputs and advice the application testing.

There are three different types of testing Alpha testing, Beta testing and Acceptance testing.

1. Alpha testing: In our project we are planning to work closely with the admin team for this testing

2. Beta Testing: In our project we are planning to release the software to all the admin teams in all the locations to test the software and raise issues if any.

3. Acceptance testing : we are planning to do the acceptance testing to decide if the system can be deployed to the customers for use.

User Stories

Product Backlog

Sprint Backlog

Acceptance Criteria

Developed and tested

Release for Customers

Accepted by Product owner/Stakeholders

**User testing -Plan**

In our project we will be doing user testing in every sprint. User stories for user acceptance testing (UAT) are writing in regular sprints. The first round of UAT testing is done during the feature creation. We will be cloning these UAT user stories to execute during the stabilization sprints. During this second UAT testing we will be validating the final integrated code to see if all the functionalities are working as expected before the software is deployed at customers site.

**Project Plan - Timelines**

To successfully accomplish that project, planning is pretty much important. To implement the proposed admission processing system project takes about 8 to 9 months of time. The proposed ADP system project includes the following six phases

*Requirements Gathering*:

During the requirements gathering phase, functional and nonfunctional requirements should be assessed well in advance i.e. the initial stage of the project. During this phase, documenting the existing systems and discussion with the stakeholders are the main tasks which are important for the further steps of the project

*Architectural Design*:

The architectural design of the ADP system project includes database designing, Software design, interface design, specifications of the system and documentation are key tasks to be completed. The estimated time frame for the architecture design phase is approximately 3 weeks

*Development*:

One of the most critical phases in the development phase in the ADP project. Along with developing the code, writing the unit test cases is the most important job during the development phase. Development phase takes longer than other phases of the ADP project. This project phase takes around 17 weeks of time.

*Testing*:

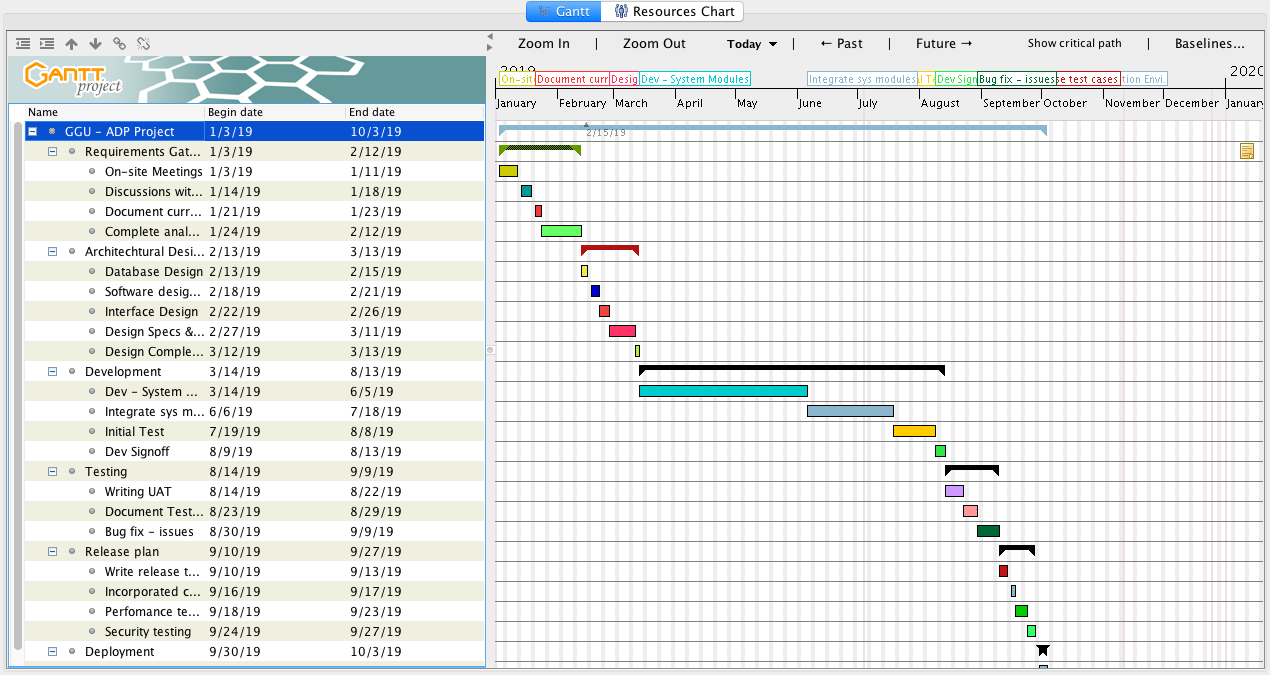
Testing is another equally important face getting the ADP development project process. During this phase, Writing UAT test cases, Documenting test cases and fixing the issues takes the equal amount of time. We have planned to allocate over three weeks of time for the testing of the entire ADP system.

*Release plan*:

Release plan comprises of three levels. Writing release test cases for all teachers, Incorporating the changes Performing Security testing and running performance tests are crucial. Release plan takes over two weeks of time during the project plan.

***Deployment*:**

Deployment is the final stage of the project. After of the rigorous testing, the ADP system will be deployed in the production environment.



**Version Controlling**

A version control system will be incorporated for entire ADP's software development life cycle. GIT is one of the robust version controlling system available in the market. GIT is user-friendly and code merge conflicts can be easily traceable in the event of multiple branch merging. A well-defined component versioning, System version, and System release policy will be maintained to avoid duplication and scrambling of the code.

*Version controlling architecture*

GIT   
Central Repository

SIT  
Environment

**PROD  
Environment**

Local Repo 1

Local Repo 2

Local Repo 3

Push

Pull

Push

Work station 1

Work station 2

Work station 3

Commit

Update

Commit

Update

Commit

Update

Deploy

Deploy

ADP System Version Controlling

**Future Plan**

The entire ADP system's function is to provide visualized metrics on Student admission by comparing with the previous years/semesters data. Based on the customers' (Golden Gate University) feedback, enhancements and modification will be taken care of the ADP system. As the usability and reliability of the system grow, mobile application will also be developed for the ease of accessibility.

**References:**

Sherman, K. (December, 2018). Testing in Agile: The Sprint Is Too Short!. Retrieved from https://www.testingexcellence.com/testing-agile-sprint-short/

Young, N. (2017, July 27). The Importance of Dashboards. Retrieved from https://www.cpaglobal.com/cpa-global-blog/the-importance-of-dashboards

Sergeev, A. (2018, June 22). Who Uses Extreme Programming. Retrieved from https://hygger.io/blog/a-brief-introduction-to-extreme-programming/