**Senior Design Report**

**Aquila**

**Process Management System**

Document Prepared By:

Bryan Bravo

Barry Boayes

Jessica Palacios

Jurelly Tello

Sasha Cisneros

Advisor:

Dr. Chengyu Sun

UAS Liaisons:

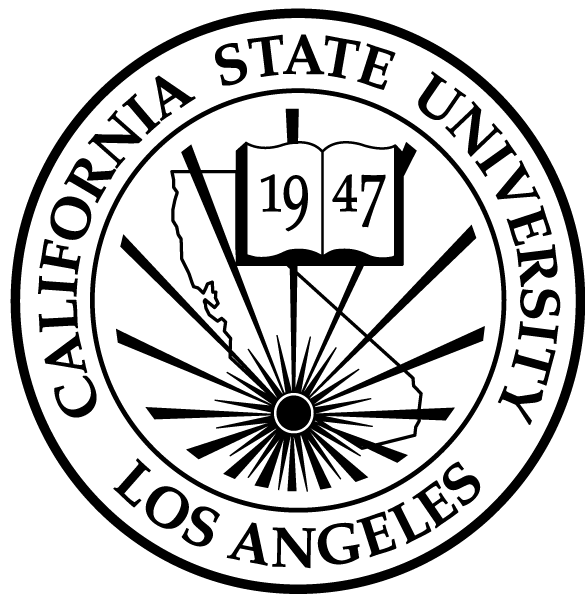
Alma Sahagun

Ernesto Argumaniz

Lynne Duong

Maria Nunes

11 May 2018



CALIFORNIA STATE UNIVERSITY LOS ANGELES

Los Angeles, California

**Table of Contents**

1. **Introduction**
2. **Related Works and Technologies**
3. **System Architecture**
4. **Results and Conclusion**
5. **References**
6. **Introduction**

The sponsor of Aquila was California State University, Los Angeles University Auxiliary Services (UAS). UAS is nonprofit organization dedicated to benefit the welfare of student, faculty, and staff. Some of the main operations of UAS is to: administer contract and grants, commercial enterprises, externally funded project, financial support services and many more. The focus of our project was the pre-award process under administration of contract and grants.

The general form of how the pre-award process works is by an applicant filling out an Intake form and creating a draft of a budget. Once those forms are completed by the applicant and reviewed by UAS this then becomes the beginning of a proposal. The applicant will then schedule a face to face meeting via email with UAS. During this meeting UAS and applicant will create a timeline. The timeline is an important factor because this is when deadlines need to be met for other forms required. Once everything is completed in the timeline then the proposal is considered completed. UAS will go over everything and if approved by them, the application will get delivered to a funding agency. It will take 6 months to a year to find out whether the proposal got approved or not by the funding agency.

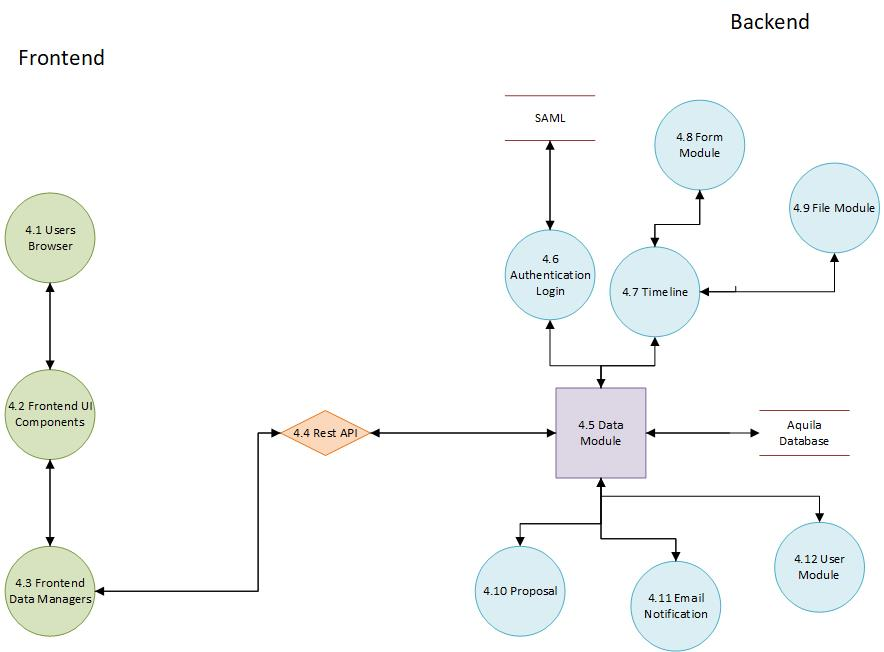
Currently the pre-award process has many limitations. One of the issues is that everything is done on physical paper. This can be problematic because it may be tedious when UAS gets audited. Another issue is that UAS may be uninformed as to the state of the project. UAS and the applicant must be in constant communication otherwise it is easy to lose track of the current state of a project.. Since physical paper is being moved around it is more susceptible to getting lost, leading to deadlines being forgotten, and causing the entire proposal to get delayed.

We decided to make a process management tool in the form of a web application. This way it is easy for the applicants and analyst to navigate through the application from any device with a browser. The applicant will be allowed to complete forms and review submitted forms. The analyst will receive email notifications on submitted forms so they can go on the site to review them. The applicant will also receive email notifications pertaining to their proposal. Using one website for both users will make it easier for both the users and the developers.

1. **Related Works and Technologies**

The Aquila application was developed having the backend and frontend completely separate. The backend was tasked to build a RESTful API and used Java as the main programming language with the use of technologies such as Eclipse as the main IDE, Spring Framework to develop controllers, Maven for building, MySQL for the database, Hibernate and JPA for helping write into the database, and Postman for testing. The backend used techniques such as Object Relational Mapping for designing objects that relate to one another in a meaningful way to be stored into the database. The backend followed architectural constraints of a RESTful API which are client and server, stateless, support caching, uniformly accessible, and layered. The frontend was tasked to build a Single Page Web Application and used Javascript as the main programming language with the use of technologies such as Visual Studio Code as the main code editor, Angular 4 framework for the client side, and PrimeNG library for UI components. The both teams used git and github for version control.

1. **System Architecture**



Aquila was built having the backend and frontend completely separate. This allows them to be developed independent of each other. This was decided to be able to manipulate data to certain endpoints to do CRUD requests. This way will also allow one team of the project to wait for the other to complete its task to continue working.

4.1 Users Browser

The user’s browser is where a majority of data is either requested or modified and saved.

4.2 Front End UI components

Large portions of the application are broken up into their own respective components. These components all have their encapsulated logic, display and styling.

4.3 Front End Data Managers

These components have data injected into them via Angular services. This paradigm of dependency injection is also present in our server implementation via Inversion of Control.

4.4 RESTful API - RESTful API is the design paradigm we are utilizing to have our User Interface communicate with the server via HTTP and JSON representation of data.

4.5 Data Module - The data module is used to take information from the frontend and send it to the correct module based on the controller the information came from. Based on the controller method, the data module is used to send out information to the frontend.

4.6 Authentication/Login Module - The login module is used to login and authenticate any of the users of our application. We will be using SAML to authenticate any of the CSULA account in addition to our internal database.

4.7 Timeline Module - Arguably the largest part of the application. The timeline modules will keep track of the users process on a proposal. Handles the uploading of files and creation of forms.

4.8 Form Module - The form module will hold all forms needed for the UAS process. These forms will be used in every application submitted to UAS.

4.9 File Module - The file module will contain any of the forms that will be submitted as a file. These forms include the budget file.

4.10 Proposal Module - The proposal module is where each proposal for a project is created. The forms and files are all included on the proposal module. The propose will take the user to the any of the forms they will need to fill out.

4.11Email Notifications Module - The email notification module will be responsible of notifying all types of users of notifications in their best interest.

4.12 User Module - Most users will be created via authentication with SAML. There are exceptions such as UAS analysts being created.

1. **Results and Conclusions**

We implemented quite a couple of features in our application for UAS. We were able to implement the role different types of users, their contact information and the different privileges they have for reading and writing to/from the database. Our team was able to map UAS’s pre-award workflow process into a digital format in a proficient matter. For example, we were able to map all the forms that were given to us into web forms. Many of these forms were very rigorous, having 30+ fields. Despite this, Aquila implements these forms with the idea of making it as straight forward for the user as possible. We delegated the responsibility of assigning forms to proposals, to the UAS analyst.swa

The most complex feature we implemented into Aquila is located in the timeline component of the proposal. Each proposal has a timeline which sets the deadlines for required forms and external documents. These external documents are represented by files that can be uploaded and downloaded. We broke up the timeline into stages that can be created, edited, deleted, and reordered. This allowed us to make break down a proposal into a linear process, which in turn allows all parties associated with a proposal to know the current state of the project and what deadlines they must adhere to. Despite completing most of the pre-award path there are a few implementations that are still necessary for Aquila.

These include the integration of SAML and digital signatures, modeling the Approval Form from the Pre-Award Process, and implementing the Post-Award Process. With SAML, users would be able to login to Aquila with their Cal State LA credentials, avoiding the process of creating new logins when unnecessary. Also, one of the main features that UAS desires for this application is digital signatures. While we were unable to integrate any digital signature API to Aquila, we did gather a great amount of information pertaining to this feature and companies that offer digital signature API integration for future implementation. Last, we did not have enough time to model the Approval Form that is included in the Pre-Award Process, or start implementing the Post-Award Process for Aquila, which is the process that takes place after a proposal has been completed and accepted. The Approval Form is apart of the last step before the submission of a final proposal, as it takes all information from the entire proposal and final University signatures.

1. **References**

The Aquila SRS and SRD were both references heavily in the creation of this document.