

# System Architecture Document

Kenan Karavoussanos

Shaylin Pillay

Preshen Goobiah

Marc Karp

October 8, 2018

## Contents

<b>1</b>	<b>User Stories</b>	<b>3</b>
1.1	PGO . . . . .	3
1.2	Supervisor . . . . .	3
1.3	PGC . . . . .	4
1.4	User Story Points . . . . .	4
<b>2</b>	<b>Documentation Deliverables</b>	<b>4</b>
<b>3</b>	<b>Task Backlog</b>	<b>5</b>
3.1	Documentation Backlog . . . . .	5
3.2	Product Backlog . . . . .	6
<b>4</b>	<b>Sprint Description</b>	<b>7</b>
<b>5</b>	<b>Sprint Plan</b>	<b>8</b>
5.1	Sprint 1: Requirements . . . . .	8
5.1.1	Sprint Retrospective . . . . .	8
5.2	Sprint 2: System Design . . . . .	9
5.2.1	Sprint Retrospective . . . . .	9
5.3	Sprint 3: System Implementation . . . . .	10
5.3.1	Sprint Retrospective . . . . .	10

# 1 User Stories

This section describes the system features from a user perspective. These user stories drive the development as their completion verifies that system requirements have been met from the user perspective. The user stories describe the feature a user wants and why they want it. Each user stories are assigned an ID for tracking and sprint planning purposes. Features that are used by multiple users have been assigned a single ID with each reason being described separately by each user. Additionally, each user story is assigned a number of story points. These points describe the overall effort required to understand, research and implement the user story feature.

## 1.1 PGO

User Story ID	I want to	So that I can
1	create an application	begin the the approval process.
2	add documents to the application	store all documents in a single repository
3	view an application	verify that application creation was successful
4	view all applications in progress	check up on applications.
5	be notified once a PGC has made a decision	upload the decision to SIMS
6	notify supervisors and the PGC	receive decisions faster.
7	view all completed applications	review previous applications.
9	login to my account	access my work exclusively.

## 1.2 Supervisor

User Story ID	I want to	So that I can
6	be notified by the PGO	be reminded to view an application.
3	view an application	make a decision on the application
8	load my decision onto the system	give the applicant my feedback.
9	login to my account	access my work exclusively.

### 1.3 PGC

User Story ID	I want to	So that I can
6	be notified once a supervisor has made a decision.	be reminded to view an application.
3	view an application	verify the decision.
8	load my decision onto the system	confirm or reject the supervisor's decision.
9	login to my account	access my work exclusively.

### 1.4 User Story Points

This section describes the story points for each user story. Story points are a relative measure of overall effort required to implement the user story feature. A baseline user story is selected as a reference with assigned score of 1. Other features are assigned a score relative to the amount of work required for the base feature i.e twice as much work will yield a score of 2.

User Story ID	Story Points
1	8
2	4
3	4
4	4
5	1
6	1
7	4
8	1
9	2

## 2 Documentation Deliverables

This section lists the document deliverables for the Project and a description of the effort required to deliver them. An effort score is assigned to each deliverable in a similar manner to the user story classification. This approach

was chosen to reflect the time spent by the team on tasks that do not contribute to a user story.

Deliverable ID	Deliverable Name	Effort Score
1	Software Requirements Specification	16
2	Software Architecture Document	32
3	Sprint Planning Document	8
4	Sprint Retrospective	2
5	Project Overview	1

### 3 Task Backlog

This section is a list of all tasks to be completed for the completion of the project. This includes tasks that move towards completing a user story as well as tasks involving project management, research and documentation. The documentation tasks will be given an effort score where one point is the equivalent effort for a story point. This approach was chosen so as to reflect the efforts of non-development team members.

#### 3.1 Documentation Backlog

This section describes the tasks required to complete the documentation deliverables for the Project.

Task ID	Task Description	Deliverable
1	Meet with PGO	1
2	Meet with PGC	1
3	Create LaTeX Template for SRS	1
4	System Feature Analysis and Decomposition	1
5	Author SRS	1
6	Create document structure and introduction for Software Architecture Document	2
7	Author Use Case View	2
8	Author Component View	2
9	Author Process View	2
10	Author Database View	2
11	Author Design View	2
12	Compile Software Architecture Document	2
13	Author Unit Test Description	2
14	Author Performance Testing Description	2
15	Compile Sprint Plan	3
16	Compile Sprint Retrospective	4
17	Compile Project Overview	5

### 3.2 Product Backlog

This section describes the tasks required to deliver the System Prototype:

Task ID	Task Description	User Stories
18	Create Application Model	1
19	Create Application View	1
20	Create Application Controller	1
21	Create Application Service	1,4
22	Set up Azure SQL DB and Blob Storage servers	2
23	Create View Application Page	4,7
24	Create Email Notification Service	5,6
25	Create Supervisor and PGC Views	8
26	Create Login Page	9
27	Create Authentication Service	9

## 4 Sprint Description

The following table provides an overview of each sprint undertaken by the development team. The Sprints are arranged around deliverables for the final project i.e System Documentation and The System Prototype.

Sprint Title	Sprint Description	Start-End Date	Deliverable
Requirements	The requirements elicitation phase including domain knowledge research and client meetings.	17 August - 31 August	System Requirements Specification
System Design	The design and documentation of the system prototype.	7 September - 7 October	System Documentation
System Implementation	The development of the system prototype.	21 September - 7 October	System Prototype

## 5 Sprint Plan

### 5.1 Sprint 1: Requirements

Task ID	Assignee			
	Kenan	Marc	Preshen	Shaylin
1	✓	✓	✓	✓
2	✓	✓	✓	✓
3				✓
4	✓	✓	✓	
5	✓			✓

#### 5.1.1 Sprint Retrospective

This sprint a lot of time was spent meeting with various stakeholders to determine the requirements of the System. Due to the sufficient note taking by our team members we were quickly able to compile and verify the list of requirements with our stakeholders. This set us up for a strong start for the next sprint and prevented us from designing unnecessary components to the system.



## 5.2 Sprint 2: System Design

Task ID	Assignee			
	Kenan	Marc	Preshen	Shaylin
6	✓			
7				✓
8	✓			
9		✓		
10				✓
11	✓		✓	
12	✓			✓
13	✓			✓
14	✓		✓	
15	✓			
16	✓	✓	✓	✓
17	✓			✓

### 5.2.1 Sprint Retrospective

The assigning of tasks this sprint was difficult due to time constraints of the members. The dependencies between tasks prevented parallel working and it was difficult to coordinate the schedules of others. However, due to good communication and meeting practices the mistakes and inconsistencies between tasks were corrected efficiently. The process of mapping our requirements into a fully fledged design was relatively smooth due to the team's experience with web application development. However, documenting the design in such detail was an area where the team was not as experienced, as such a lot of time was spent learning about the documentation procedures. In conclusion, the sprint was marked with various issues but good communication and our previous experience minimized the impact of said issues.

### 5.3 Sprint 3: System Implementation

Task ID	Assignee			
	Kenan	Marc	Preshen	Shaylin
18		✓	✓	
19	✓	✓	✓	✓
20		✓	✓	
21		✓	✓	
22		✓	✓	
23	✓	✓	✓	✓
24		✓	✓	
25		✓	✓	
26		✓	✓	
27	✓	✓	✓	

#### 5.3.1 Sprint Retrospective

The document file storage was a big design and architecture concern throughout this sprint. Initially, storing the files in the SQL Database caused unacceptable performance issues. As such, it was decided upon to store the files in an Azure Blob Storage container and store the URL to the file within the SQL Database. Additionally, Azure Blob Storage uses Geo-Redundant storage which automatically provides safe back up for files. It is important to note for future projects that storage of binary data in relational databases causes impacts performance significantly. For future development, all CSS and JS files should be bundled and minified to improve the load time of the web application.