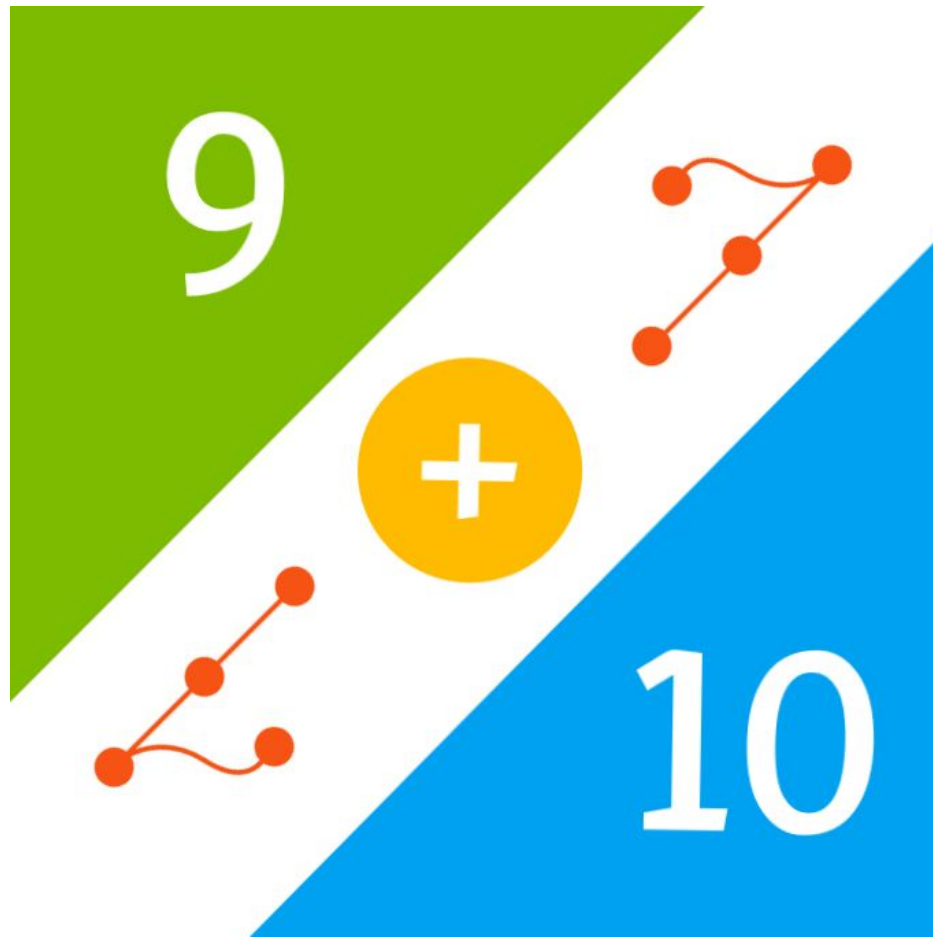


# CSCC01 Introduction to Software Engineering

Project Deliverable 3 - Project Planning and Execution



*Team 9 + 10*

**Shruti Mistry**  
**Dennis Tismenko**  
**Abbas Rattansi**  
**Qi Lin**  
**Raya Farhadi**

<b>Tasks</b>	<b>3</b>
<b>Sprint Backlog - Week 1</b>	<b>6</b>
Sprint Plan	6
Sprint Report	7
Burndown Chart	7
<b>Sprint Backlog - Week 2</b>	<b>8</b>
Sprint Plan	8
Sprint Report	8
Burndown Chart	8

## Tasks

User Story	Task	Dependency	Story Points	Description
1	1a. Create the login page		4	Create and implement the frontend and backend for the login functionality. Backend should verify password against a SHA-256 hash (and salt). Once logged in, user should be stored in a 'session'.
1	1b. Create the initial user account		1	Create the 'admin' account that will be used for development. Using Restdb.io for the database, create an account in the 'Users' table.
1	1c. Create the permissions columns		3	Create Permission collection and set up permission structure for the future usage. Be Careful, please write something that can be expand and easy to track.
1	1d. Create account page with basic user information		2	Create account page with basic user information. It should have a drop down box that are use for role selection. In that drop down box if the user selects the agency role, it should display a text

				field that allow user to input agency name and few selective box for service. Those new component should only be visible for the agency selection.
1	1e. Backend implementation of creating a user account	1d	1	Set up correct user model and associated it with the creating page data. Construct JSON Object and call a post request for the database.
2	2a. Create user management tab with search functionality		4	Create a backend and frontend feature that present a search button which are calling some get request to restDB and present a list of relevant data for the user.
2	2b. Create an editable table for the templates page		3	Create expandable column header with submitting and cancel button on it. We want to have empty table that does not have any columns, there is two buttons beside the table that allow user to click. The first button will create the new column in the table which by default is name by " new table column". The second button will delete the selected column. We should also allow user to double click the column header, which will enable user to change the column

				names.
2	2c. Create an upload button for Excel spreadsheets (into the template editor)		3	Be careful, we should also linked the data to the data model. Be consistent.
2	2d. Implement backend functionality for Excel upload feature		3	Using Apache Poi library to implement the excel upload feature. Please try to convert it to some readable object first, so that later on we can just use it for the submission button.
2	2e. Preload all templates provided by client (into database)		2	Besure to update with the new requirement from client meeting. It is also important for match with the new database model.
2	2f. Create database model for storing templates		3	Create a expandable database model. Note: besure to communicate with other teammate, so that everyone can have good understanding on the new database model.
2	2g. Populate database with submitted template columns		3	We want to put column names of the new table into the right service(template set). This will be depending on the service selection which locating in the template creation page.
3	3a. Create 'template browsing' tab		2	In term of UI, we want to have a fresh and clean UI with some good instruction list beside the buttons.
3	3b. Implement template selector		2	We want to use the

	feature (for browser and editor)			search feature from 2a) for the browsing feature, be sure to incorporate with the new database model.
3	3c. Retrieve data about currently logged in User		2	This can be implemented by calling post request and store info into the current session data model.

## Sprint Backlog - Week 1

### Sprint Plan

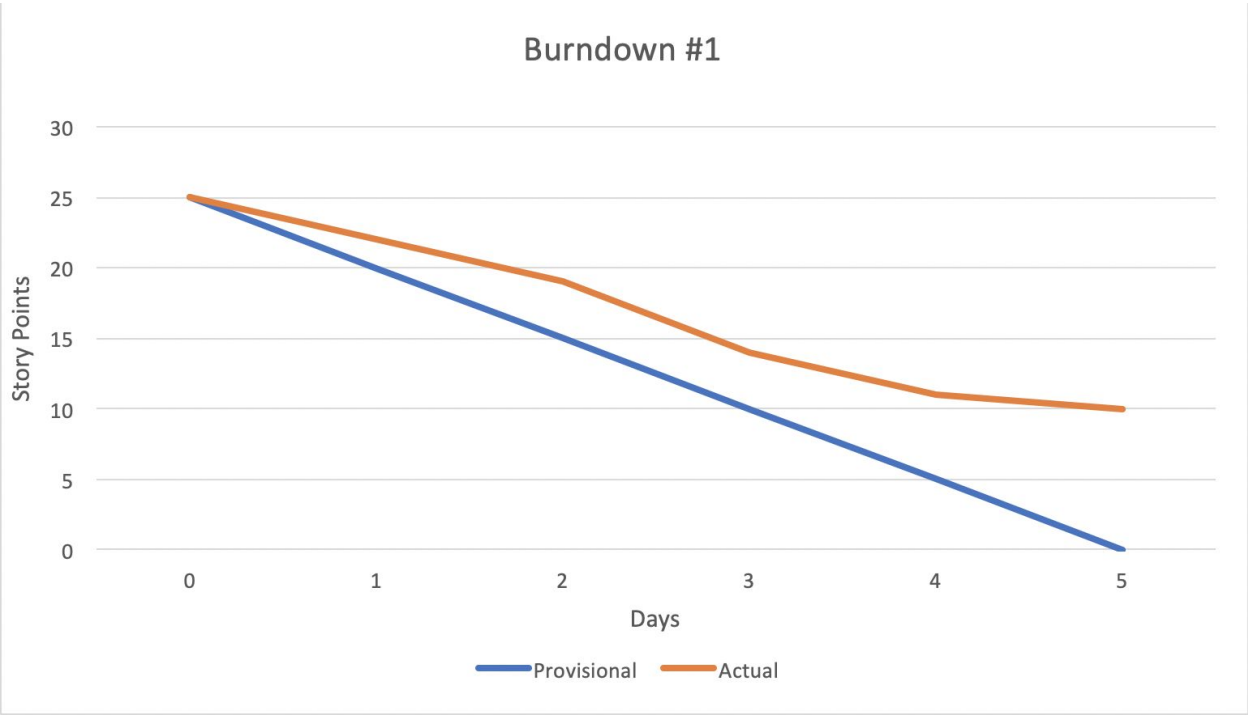
Sprint 1 Plan	1 hour per developer = 5*5*1 = 25								
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5	
1 1a			4	D:1	D:1	D:1	D:1		
1 1b			1		D:1				
1 1c			1		S:1				
1 1d			2	Q:1	Q:1				
1 1e		1d	1			A:1			
2 2c			3	Q:1	Q:1			Q:1	
2 2d			3		D:1	D:1	D:1		

Sprint Report

Sprint 1									
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5	
1	1a		4	D:3	D:1				
1	1b		1			D:1			
1	1c		1			S:1			
1	1d		2		Q:2				
1	1e	1d	1				A:1		
2	2c		3			Q:3			
2	2d		3				D:2	D:1	

Burndown Chart

Burndown #1	0	1	2	3	4	5
Provisonal	25	24	21	15	12	10
Acctual	25	22	19	14	11	10



Sprint Backlog - Week 2

Sprint Plan

Sprint 2 Plan	1 hour per developer = 5*5*1 = 25								
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5	
	2 2a		4 R:1	R:1		R:1	R:1		
	2 2g		3	S:1		S:1	S:1		
	2 2h		3 Q:1	Q:1		Q:1			
	3 3a		2	D:1		D:1			
	3 3c		2	A:1	A:1				

Sprint Report

Sprint 2									
User Story	Tasks	Dependencies	Story Points	1	2	3	4	5	
	2 2a		4 R:2						
	2 2g		3	S:1		S:2			
	2 2h		3			Q:2			
	3 3a		2	D:1					
	3 3c		2	A:1					

Burndown Chart

Burndown #2	0	1	2	3	4	5
Provisonal	25	23	18	17	13	11
Acctual	25	23	20	20	16	16



