COMP1531 | T09B / H17B

Week 9

william.huynh3@unsw.edu.au

UNSW SYDNEY

Learning objectives

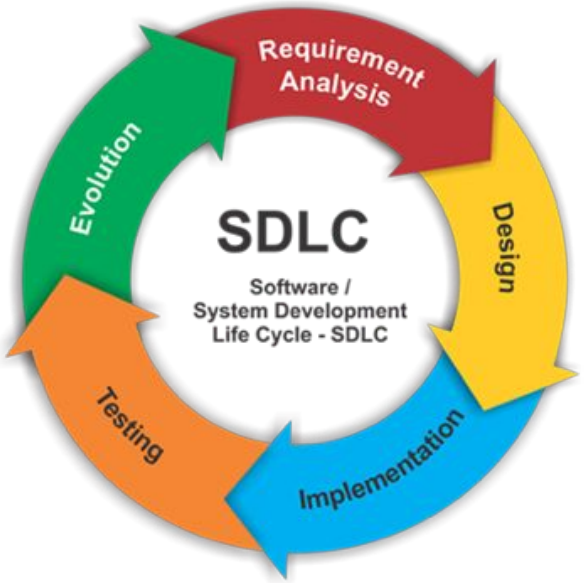
- **It3 Requirements Report** [15 % Weighting]
  - Elicitation (2 marks)
  - Use Cases (3 mark)
  - Validation (1 mark)
  - Interface Design (2 marks)
  - State Diagrams (2 marks)

UNSW SYDNEY

Requirements Report: Why ?????

The requirements report is essentially your plan for an Iteration 4

And it is the **"Evolution"** stage of the SDLC



1. Reading the Spec
2. Writing test + stub functions
3. Implementing your functions
4. Continuous Integration *(passing the pipeline)*
5. New Iterations!

Evolution: Planning for a imaginary Iteration 4

1. Elicitation → Gathering requirements through Questionnaires & Interviews

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features

5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY

Elicitation: Questionnaire and Interview

The Elicitation process allows developers to discover problems from existing software through the use of questionnaires and interviews

Marking Criteria:

- [0.5 Marks] There are at least 4 questions
- [0.5 Marks] There are at least 2 interviews with the responses recorded
- [1 Marks] The interview focuses on finding problems, rather than finding solutions

Q: What features do you think are lacking in the current Teams software you use?

A: I wish there was a voice chat feature because I want to etc...

Elicitation is worth 2 marks of the Requirements Report

UNSW SYDNEY

Task

1. Create a Google Doc 📝

2. Write down 4 questions to ask users 🧐
   a. (1 question per team member)

3. Go around to another group and interview them 🦜

   Remember to record their:
   a. Name & zID 🙇
   b. Email 📧
   c. Home Address 🏠
   d. Credit card + expiry date + 3 digits on the back 🤑

Marking Criteria:

- [0.5 Marks] There are at least 4 questions
- [0.5 Marks] There are at least 2 interviews with the responses recorded
- [1 Marks] The interview focuses on finding problems, rather than finding solutions

Evolution: Planning for a imaginary Iteration 4

1. Elicitation → ~~Gathering requirements through Questionnaires & Interviews~~

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features

5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY

Use Cases: Analyse our interview responses

Once we have our interview responses, we need to consolidate them into actionable information.

This means, for each response we need to generate:

User Stories & Acceptance Criteria
Use Cases

Marking Criteria:

- [1 Marks] Responses have been translated into user stories
- [1 Marks] User stories contain Acceptance Criteria
- [1 Marks] User stories have been translated into Use Cases

Use Cases are worth 3 marks of the Requirements Report

UNSW SYDNEY

User Stories & Acceptance Criteria: User Stories

User Stories **formalise responses** taken from an interview

They consist of a User Story and an Acceptance Criteria

Response

It would be cool if I could chat with my work colleagues after work over a private channel so the boss won't find out lmaooo xD

→

User Story

As an office worker, I want the ability to create private channels so I can chat with colleagues privately outside of work.

As a `<type of user>`, I want `<some goal>` so that `<some reason>`

User Stories & Acceptance Criteria: Acceptance Criteria

Once you have a user story, you should create the acceptance criteria for that story.

The acceptance criteria is a recipe of actions that you can take to '*accept/achieve*' what that user wants.

As an office worker, I want the ability to create private channels so I can chat with colleagues privately outside of work.

- Once a user registers an account, they can create a channel by pressing the 'plus' button on home page
- The user is then prompted to enter the name of their channel *(max 50 characters)*
- The user can then select a list of friends from a drop-down menu to add to their channel
- After they can send chat messages to the channel, and all the members can see those messages, and react or reply.

Task

1. In pairs, create a user story from your responses 🧐
   a. 1 story for each response

2. In pairs, create a recipe of Acceptance Criteria for that User Story

Response

It would be cool if I could chat with my work colleagues after work over a private channel so the boss won't find out lmaooo xD

→

User Story

As an office worker, I want the ability to create private channels so I can chat with colleagues privately outside of work.

As a `<type of user>`, I want `<some goal>` so that `<some reason>`

As an office worker, I want the ability to create private channels so I can chat with colleagues privately outside of work.

→

- Once a user registers an account, they can create a channel by pressing the 'plus' button on home page
- The user is then prompted to enter the name of their channel *(max 50 characters)*
- The user can then select a list of friends from a drop-down menu to add to their channel
- After they can send chat messages to the channel, and all the members can see those messages, and react or reply.

Marking Criteria:

- [1 Marks] Responses have been translated into user stories
- [1 Marks] Use cases are consistent with the user stories
- [1 Marks] Acceptance Criteria are written like a recipe

Use Cases:

Now that we have User Stories with Acceptance Criteria,
We need to create Use Cases

✍🏼 Use-Case List

We can provide a use case in written form.

- Step 1. ATM asks customer for pin
- Step 2. Customer enters pin
- Step 3. ATM asks bank to verify pin and account
- Step 4. Bank informs ATM of validity and balance of account
- Step 5. ATM asks customer what action they wish to take
- Step 6. Customer asks to withdraw an amount of money
- Step 7. ATM Dispenses money to customer
- Step 8. ATM informs bank of withdrawal

Task

1. In pairs, create a use case for each user story
   a. 1 use case per user story

✍🏼 Use-Case List

We can provide a use case in written form.

- Step 1. ATM asks customer for pin
- Step 2. Customer enters pin
- Step 3. ATM asks bank to verify pin and account
- Step 4. Bank informs ATM of validity and balance of account
- Step 5. ATM asks customer what action they wish to take
- Step 6. Customer asks to withdraw an amount of money
- Step 7. ATM Dispenses money to customer
- Step 8. ATM informs bank of withdrawal

Marking Criteria:

- [1 Marks] Responses have been translated into user stories
- [1 Marks] Use cases are consistent with the user stories
- [1 Marks] Acceptance Criteria are written like a recipe

Evolution: Planning for a imaginary Iteration 4

1. Elicitation → Gathering requirements through Questionnaires & Interviews

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features

5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY

Evolution: Planning for a imaginary Iteration 4

1. Elicitation → Gathering requirements through Questionnaires & Interviews

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features

5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY

Interface Design

Now that we know the details for the features we want to implement,

We need to create an interface for those features.

Variable Interface

Variable name	Type
named exactly **email**	string
has suffix **id**	integer
contains substring **password**	string
named exactly **message**	string
named exactly **start**	integer

Function Interface

Name & Description	HTTP Method	Data Types	Exceptions
`auth/login/v3` Given a registered user's `email` and `password`, returns their `authUserId` value.	POST	**Body Parameters:** `( email, password )` **Return type if no error:** `{ token, authUserId }`	**400 Error** when any of: • `email` entered does not belong to a user • `password` is not correct

Marking Criteria:

- [1 Marks] HTTP Interface is in the correct structure
- [1 Marks] HTTP Interface is consistent with the use cases

UNSW SYDNEY

Task

1. Come up with at least 4 functions related to your new features

2. For each function generate a function interface for it
   a. Route + Description
   b. HTTP Method
   c. Data Types (parameters & returns)
   d. Exceptions

3. Create a variable interface that accounts for all your Data Types

Variable name	Type
named exactly **email**	string
has suffix **id**	integer
contains substring **password**	string
named exactly **message**	string
named exactly **start**	integer

Name & Description	HTTP Method	Data Types	Exceptions

`auth/login/v3`
 	
Given a registered user's `email` and `password`, returns their `authUserId` value.
	POST	**Body Parameters:**
		`{ email, password }`
		
		Return type if no error:
		`{ token, authUserId }`
			400 Error when any of:
			• `email` entered does not belong to a user
			• `password` is not correct

Marking Criteria:

- [1 Marks] HTTP Interface is in the correct structure
- [1 Marks] HTTP Interface is consistent with the use cases

Evolution: Planning for a imaginary Iteration 4

1. Elicitation → Gathering requirements through Questionnaires & Interviews

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features

5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY

State Diagrams

A State Diagram is a diagrammatic representation of a state.

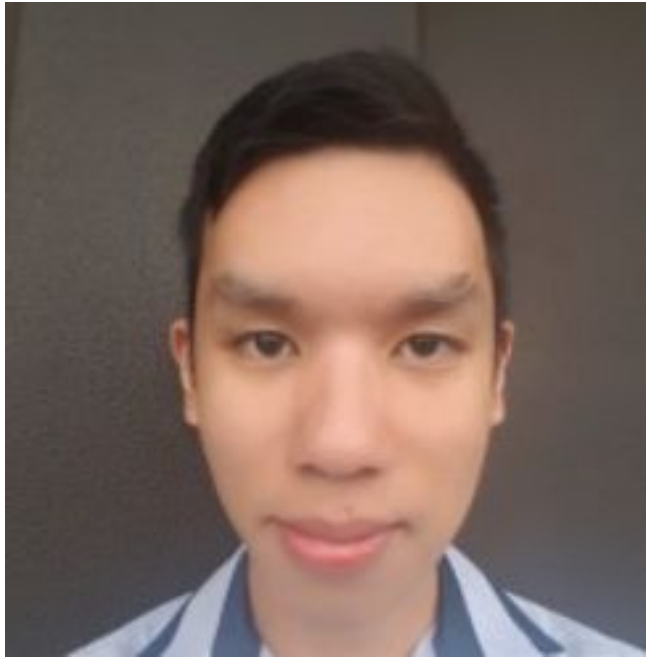
There are two symbols in a state diagram: Circles & Arrows

Circles represent states.

Arrows represents transitions

Marking Criteria:

- [1 Marks] State Diagram is in the correct structure *(circles and arrows)*
- [1 Marks] State Diagram is consistent with the use cases



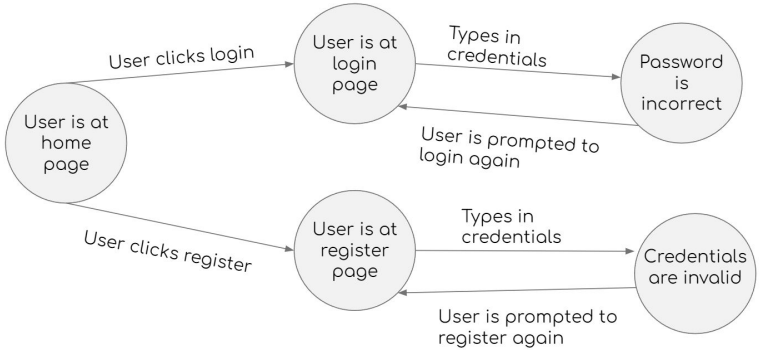
Task

1. Create a state diagram for your use cases

   a. Feel free to do a single state diagram that summarises both use cases

   OR

   b. Create two individual state diagrams for each use case



Marking Criteria:

- [1 Marks] State Diagram is in the correct structure *(circles and arrows)*
- [1 Marks] State Diagram is consistent with the use cases

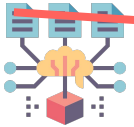
Evolution: Planning for a imaginary Iteration 4

1. Elicitation → Gathering requirements through Questionnaires & Interviews

2. Use Cases → Analyse the responses of your interview and create user stories

3. Validation → Show the user stories to interviewees, and get their feedback

4. Interface Design → Create an interface for the HTTP endpoints for your new features



5. Conceptual Modelling → Create a state diagram for one of your new features

UNSW SYDNEY