GROUP ASSESSMENT ITEM COVER SHEET

Student Numbers:	Emails:	FIRST NAMES	FAMILY / LAST NAMES
3 2 8 2 9 3 8	c3292938@uon.edu.au	Simeon	Pento
3 2 8 0 7 5 1	c3280751@uon.edu.au	William	Paterson
3 2 8 3 3 4 4	c3283344@uon.edu.au	Lachlan	McRae

Course Code

S	E	N	G	2	0	5	0

(Example)

A	B	C	D	1	2	3	4

Course Title

Web Engineering

(Example)

Intro to University

Campus of Study: _____ Callaghan _____ (eg Callaghan, Ourimbah, Port Macquarie)

Assessment Item Title: Assignment 3 - Documentation and Plan   Due Date/Time: 26/5/19

Tutorial Group (If applicable): [_____]   Word Count (If applicable): [_____]

Lecturer/Tutor Name: _____

Extension Granted: [ ] Yes   [x] No   Granted Until: [_____]

Please attach a copy of your extension approval

NB: STUDENTS MAY EXPECT THAT THIS ASSIGNMENT WILL BE RETURNED WITHIN 3 WEEKS OF THE DUE DATE OF SUBMISSION

Please tick box if applicable

[ ] *Students within the Faculty of Business and Law, Faculty of Science, Faculty of Engineering and Built Environment and the School of Nursing and Midwifery:* We verify that we have completed the online Academic Integrity Module and adhered to its principles.

[ ] *Students within the School of Education:* We understand that a minimum standard of correct referencing and academic literacy is required to pass all written assignments in the School of Education; and we have read and understood the School of Education *Course Outline Policy Supplement,* which includes important information related to assessment policies and procedures.

We declare that this assessment item is our own work unless otherwise acknowledged and is in accordance with the University's Student Academic Integrity Policy
We certify that this assessment item has not been submitted previously for academic credit in this or any other course. We certify that we have not given a copy or have shown a copy of this assessment item to another student enrolled in the course, other than members of this group.

We acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:
• Reproduce this assessment item and provide a copy to another member of the Faculty; and/or
• Communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the item on its database for the purpose of future plagiarism checking).
• Submit the assessment item to other forms of plagiarism checking.

We certify that any electronic version of this assessment item that we have submitted or will submit is identical to this paper version.

Turnitin ID:
(if applicable) [_____]

DATE STAMP HERE

Signature:	_____	Date: _____
Signature:	_____	Date: _____
Signature:	_____	Date: _____
Signature:	_____	Date: _____
Signature:	_____	Date: _____

Insert this way

SENG2050 – Assessment 3 – Documentation and Plan

Authors: Simeon Pento (c3282938), William Paterson (c3280751) & Lachlan McRae (c3283344)

Use Case Descriptions: William Paterson

Login	
MAIN ACTORS:	*Users* *Staff*
SECONDARY ACTORS:	*System*
DESCRIPTION:	The *Actor* is logged into the *System*
BASIC FLOW:	1. *Actor* enters username and password 2. *Actor* clicks login 3. *System* authenticates login information 4. *Actor* is redirected to main page
ALTERNATIVE FLOW:	4a. *System* determines if *Actor* is *Staff* *System* sets User Type as staff member 4b. *System* fails to authenticate *Actor* *Actor* is informed login information is wrong - Login page is reloaded
PRE-CONDITION	Page is loaded
POST-CONDITION	*Actor* is logged in

View Specific Report	
MAIN ACTORS:	*Users* *Staff*
SECONDARY ACTORS:	*System*
DESCRIPTION:	A specific report is displayed
BASIC FLOW:	1. *Actor* selects report 2. *System* finds report 3. Report is displayed to *Actor*
ALTERNATIVE FLOW:	2a. No report exists *System* informs *Actor* of error *Actor* is redirected to main page
PRE-CONDITION	- Page is loaded *Actor* is logged in
POST-CONDITION	A specific report is displayed

View Open Reports	
MAIN ACTORS:	• *Users* • *Staff*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	Any open reports are displayed
BASIC FLOW:	1. *System* retrieves open reports 2. Reports are displayed to *Actor* in a table
ALTERNATIVE FLOW:	1a. No reports exist • *System* informs *Actor* no reports exist
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• Open Reports are shown

View Knowledge Base	
MAIN ACTORS:	• *Users* • *Staff*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	All Completed reports are displayed
BASIC FLOW:	1. *System* retrieves completed reports 2. Reports are displayed to *Actor* in a table
ALTERNATIVE FLOW:	1a. No reports exist • *System* informs *Actor* no reports exist
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• Completed Reports are shown

Add Comment	
MAIN ACTORS:	• *Users* • *Staff*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	Allows actors to add comments to reports
BASIC FLOW:	1. *Actor* selects report 2. *Actor* clicks on add comment 3. *System* redirect to add comment page 4. *Actor* enters comment details 5. *Actor* clicks submit 6. *System* add comment to report database 7. *Actor* is redirected to main page
ALTERNATIVE FLOW:	1a. No report exists • *System* informs *Actor* of error • *Actor* is redirected to main page
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• A comment is added to a specific report

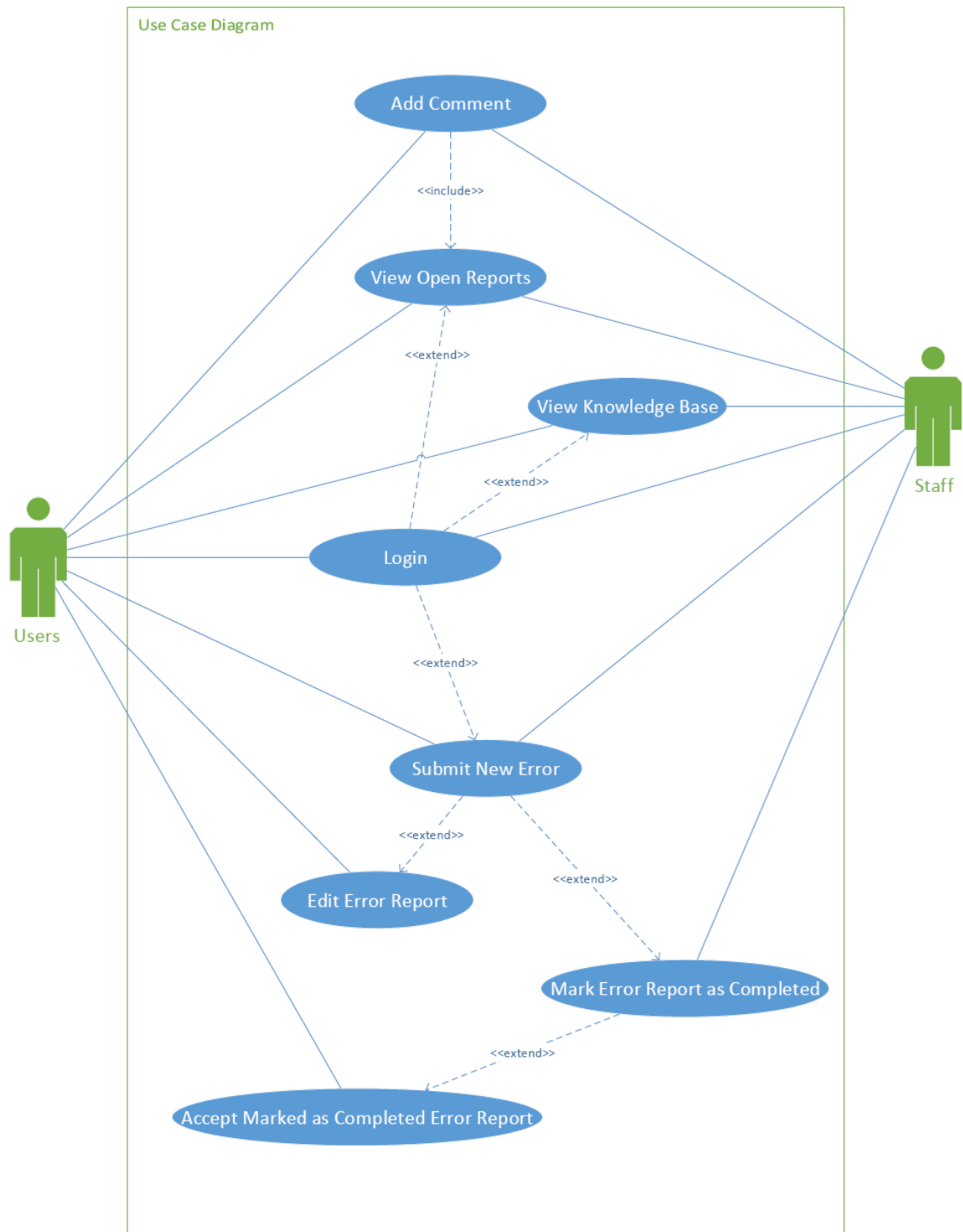| Mark Error Report as Completed	
MAIN ACTORS:	• *Staff*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	A specific report is marked as completed
BASIC FLOW:	1. *Actor* selects report 2. *Actor* marks report as completed 3. *System* updates report status 4. *Actor* is redirected to main page
ALTERNATIVE FLOW:	1a. No report exists • *System* informs *Actor* of error • *Actor* is redirected to main page
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• A specific report is marked as completed

Accept Marked as Completed Error Report	
MAIN ACTORS:	• *Users*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	Actor can accept or deny error completion
BASIC FLOW:	1. *Actor* selects report 2. *System* informs *Actor* report has been marked as completed 3. *Actor* can then click accept or deny 4. *System* updates status of report 5. *Actor* will be redirected to main page
ALTERNATIVE FLOW:	1a. No report exists • *System* informs *Actor* of error • *Actor* is redirected to main page
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• Error report is accepted or denied

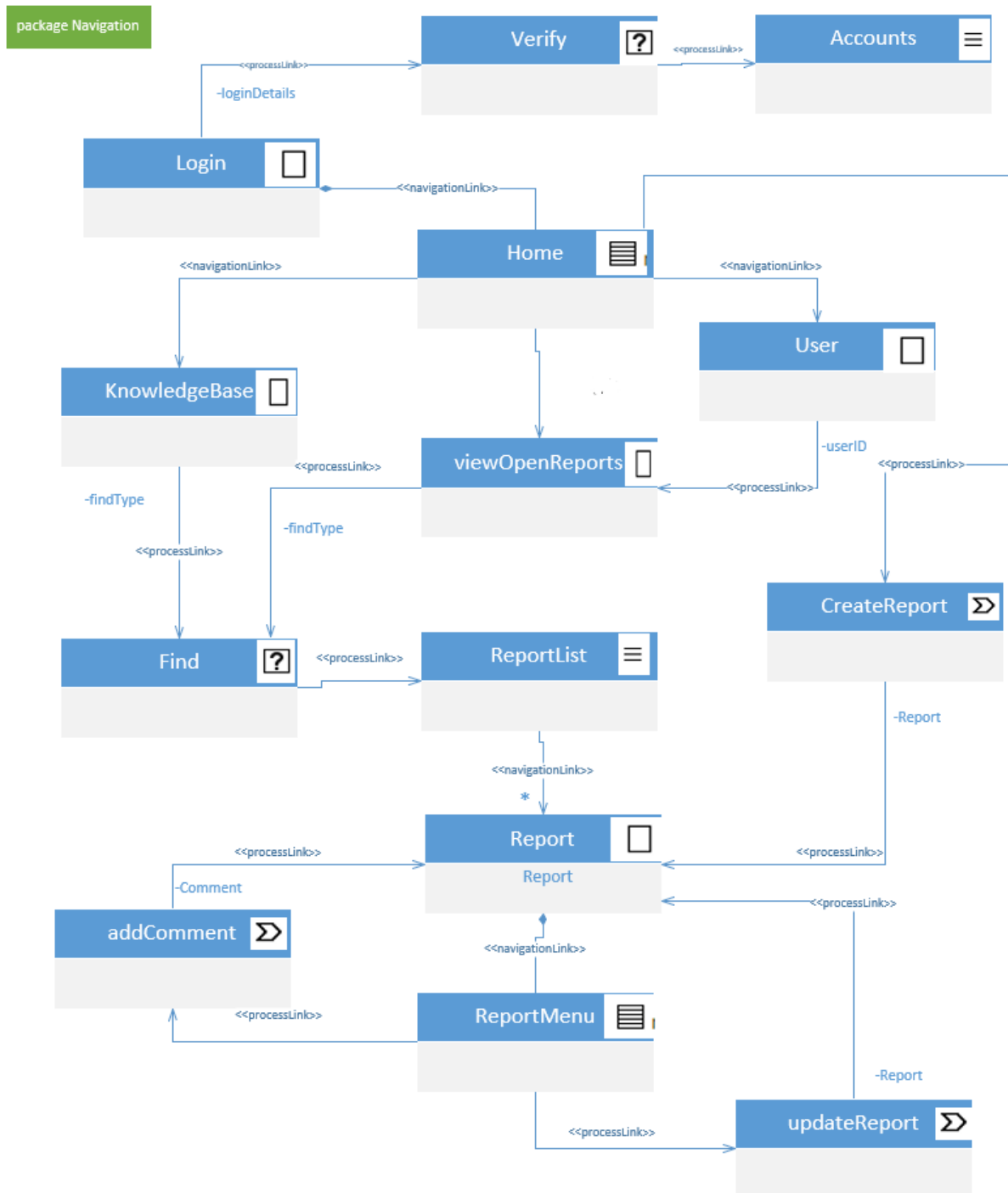| Add to Knowledge Base	
MAIN ACTORS:	• *Staff*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	Add a completed report to the knowledge base
BASIC FLOW:	1. Actor opens completed report 2. Actor clicks to add to Knowledge Base 3. System informs Actor report was added 4. Actor is redirected to main page
ALTERNATIVE FLOW:	3a. Actor clicks do not add to Knowledge Base • Actor is redirected to main page
PRE-CONDITION	• Report is marked as completed • Page is loaded • *Actor* is logged in
POST-CONDITION	• A report is added to Knowledge base

Submit New Error Report	
MAIN ACTORS:	• *Users*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	A new error report is submitted
BASIC FLOW:	1. *Actor* is asked to enter details of error 2. *Actor* clicks submit 3. *System* validates details 4. Entered details are entered into the database by the *System* 5. *System* informs *Actor* details were successful entered 6. *Actor* is redirected back to main page
ALTERNATIVE FLOW:	2a. *Actor* clicks clear • All entered details are cleared • Page is reloaded 3a. Details are invalid • *System* informs *Actor* of error • *Actor* is redirected to main page 4a. *Actor* attaches a file • *System* stores the file 5a. Error adding details to database • *System* informs *Actor* of error • *Actor* is redirected to main page
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• A new report is submitted

Edit Error Report	
MAIN ACTORS:	• *Users*
SECONDARY ACTORS:	• *System*
DESCRIPTION:	Allows Actors to edit reports
BASIC FLOW:	1. Actor edits details 2. Actors clicks submit 3. System validates details are
ALTERNATIVE FLOW:	2a. No report exists • *System* informs *Actor* of error • *Actor* is redirected to main page
PRE-CONDITION	• Page is loaded • *Actor* is logged in
POST-CONDITION	• A specific report is edited

Use Case Diagram: William Paterson

Navigation Models: Lachlan McRae

package Navigation

Verify [?] <<processLink>> Accounts [≡]

<<processLink>>
-loginDetails

Login [□] <<navigationLink>>

Home [▤]

<<navigationLink>> <<navigationLink>>

KnowledgeBase [□] User [□]

<<processLink>> -userID
viewOpenReports [□] <<processLink>>
-findType
<<processLink>> -findType

CreateReport [»]

Find [?] <<processLink>> ReportList [≡]

-Report

<<navigationLink>>
*

Report [□]
<<processLink>> Report <<processLink>>
-Comment <<processLink>>

addComment [»]

<<navigationLink>>

<<processLink>> ReportMenu [▤]

-Report

<<processLink>> updateReport [»]

Presentation Models: Lachlan McRae

View Open Reports

: Open Reports ≋

: Report [*]

: Location ≋	: picture 🔲
: Browser ≋	
: website ≋	
: Intranet	
: other Browser ≋	
: restarted ≋	: form Add Comment ●
: description ≋	: Edit Report ●

: Comment [*]
: comment [abl]

Knowledge Base

: Knowledge Base ≋

: Report [*]

: Location ≋	: picture 🔲
: Browser ≋	
: website ≋	
: Intranet ≋	
: other Browser ≋	
: restarted ≋	
: description ≋	

: Comment [*]
: comment [abl]

Content Model: Simeon Pento

User

- String : name
- Boolean : staff

Contact Database

Report

- DateTime : date
- String : title
- String : category
- int : status
- int : numOfComments
- Boolean : inKnowledge

Comment

- String : text
- DateTime : date

Files

- String : Path
- Boolean : exists
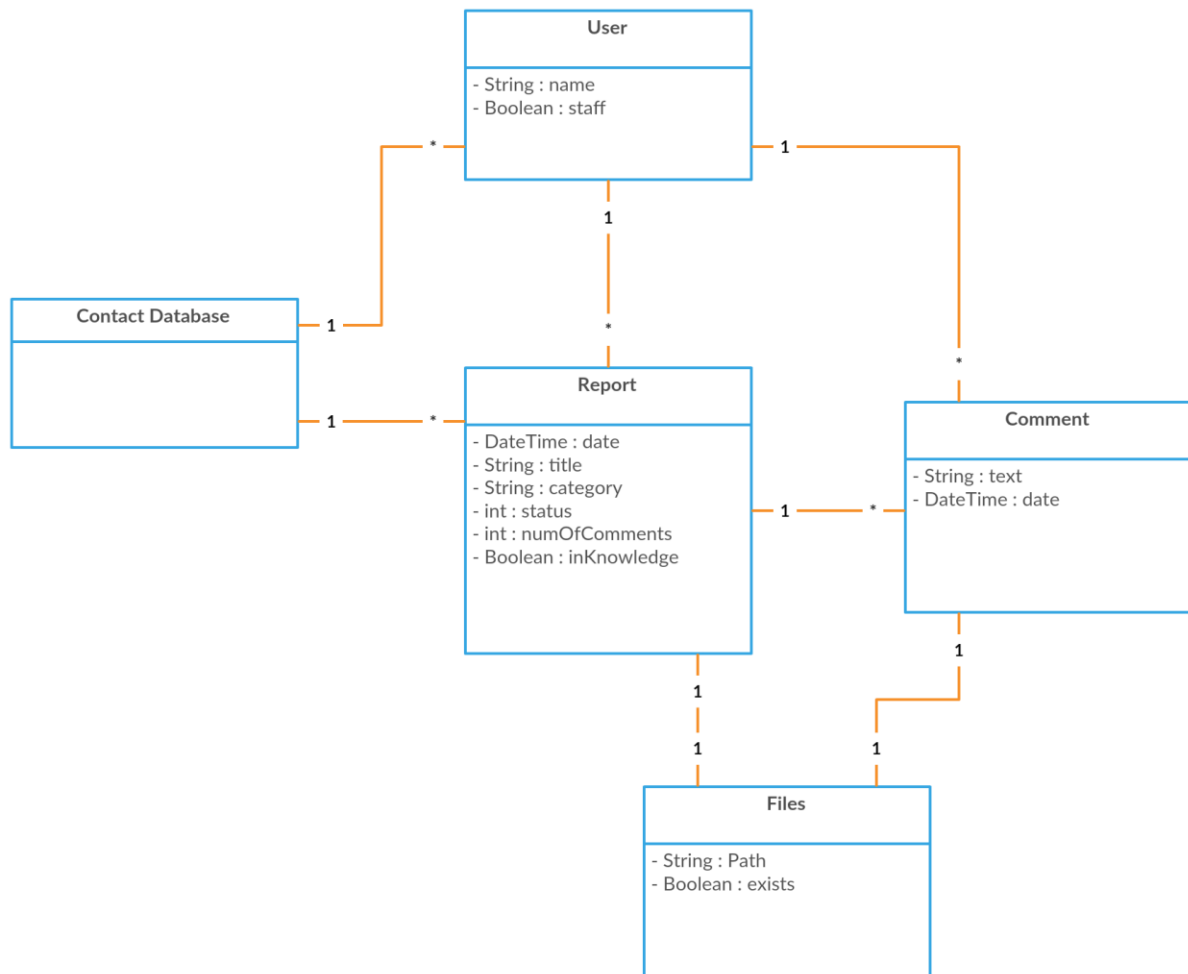
Development Plan: Simeon Pento

ID	Activities	17 Ma	19 May 2019								26 May 2019								2 Jun 2019								9
		F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	Documentation		100%																						
1.1	- Documentation	7 Days 100 %																							
2	Coding					0%																			
2.1	- SQL Database					3 Days 0 %																			
2.2	- Objects					0%																			
2.2.1	- Users.Java					1 Day 0 %																			
2.2.2	- Comment.java						1 Day																		
2.2.3	- Files.java						1 Day 0 %																		
2.2.4	- Report.java							1 Day																	
2.3	- JSPs								3 Days 0 %																
2.4	- Servlets										8 Days 0 %														
3	Debugging																		3 Days 0 %						
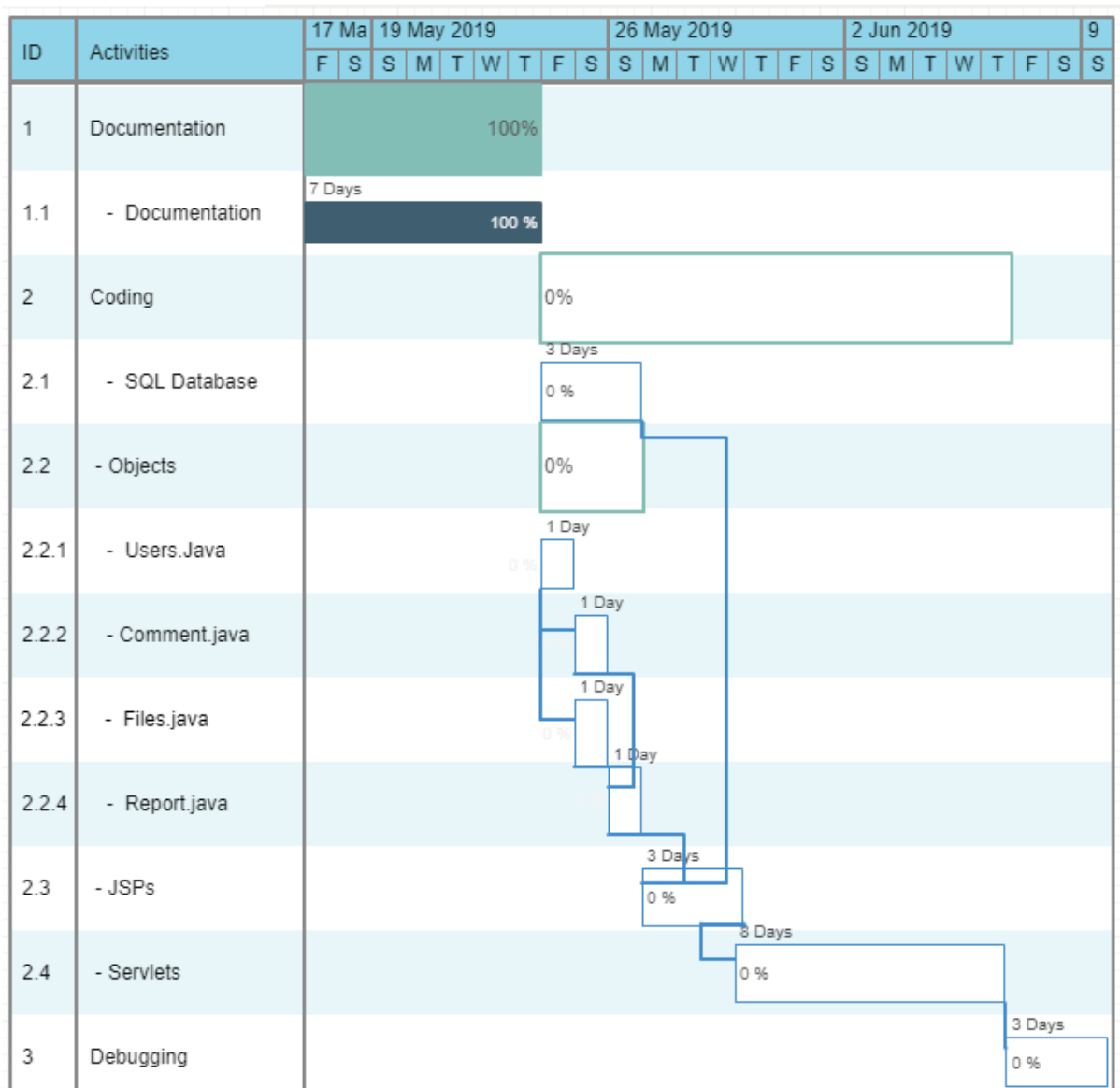
Project Breakdown:

- William:
    - SQL Database
    - Servlets
- Simeon:
    - Java Beans / Objects
    - JSPs
    - Servlets
- Lachlan
    - Servlets
    - Final debugging