

# **COM1001 Introduction to Software Engineering**

## **Iteration 2: Team Report**

### **Group 12A**

#### **Names**

Yaxin Ding

Alex Parr

Aryan.

Luke Suckling

Nathan Mitchell

Poom Hirunwiwatkul

Ramona Petre

## **Part 1: How to access the system**

### **1.1 Starting the system**

“cd project/src”

“rake run or ruby app.rb”

### **1.2 Logging in**

<u>Username</u>	<u>Password</u>
admin	admin
role1	role1
role2	role2
role3	role3
role4	role4

### **1.3 Account info**

‘admin’ is an admin account, it has access to view reports and manage accounts. Managing accounts include being able to approve accounts and delete accounts.

‘role[1-4]’ are user accounts, they have read/write access to all the bookmarks: they can comment, add tags, rate, favourite, report and create new bookmarks.

When not logged in, user can view bookmarks and view comments but cannot add tags, comment, rate, favourite, report or create new bookmarks.

## **‘Part 2: User stories**

### **2.1 Table of stories**

Description	Acceptance Criteria	Client	Est.	Implemented
-------------	---------------------	--------	------	-------------

		Priority	Effort	
<b>Bookmark System</b>				
As a user I want to be able to click on a link to a bookmark so that I can view more detailed information about it such as comments, description, date created, bookmark's url, tags, author and rating, to understand more about the bookmark.	See bookmark in list of search results Click the bookmark and see it in a new window Want to view the information where no information exists I want to see empty fields.	H	1	✓
As a user I want to be able to add a short description to a bookmark so that other users know what it is without having to open it.	View a field for each piece of information	H	2	✓
As a user, when I add a new bookmark, it should be checked so that I can make sure I'm not creating a duplicate so that the bookmark system is not inefficient	When attempting to add a duplicate the user is alerted it is a duplicate When not a duplicate successfully created	M	3	✓
<b>Search System</b>				
As a user, I want all the bookmarks to be searchable based on their tags so that I can quickly find the one I am looking for.	When I search see all bookmarks in a table Then click through to individual bookmark page	H	3	✓
As a user, I want to insert a keyword in order to search for bookmarks so that I can find the bookmarks quicker	When I search fetch bookmarks with the keywords in the title, description, comments	H	8	✓
As a user, I want to be able to sort the search results based on: rating, date created, popularity, alphabetical (author), alphabetical (tags), most relevant so that I can more easily find the one I'm looking for.	When I search I want the table to appear below the search bar and/or then sort by the chosen parameter	M	5	✓ partially

Reporting System				
As a user, I want to be able to give a short reason why I'm reporting a tag, so that the admin can see a valid reason as to why the bookmark is being reported and doesn't have to find the problem themselves	When I click on the report button, a report form pops up where a user can enter a short description why the tag has been reported	M	3	✓
Accounts				
As a user I want to be able to login so that I can have full read write access to the bookmarks.	I want to be able to enter a username, email and password and, if correct, then be taken to the search page. If incorrect, then the user is <b>not</b> taken to the user page	H	5	✓
As a user, without an account, I want to be able to read all the bookmarks, so that I can still access the company material	I want to be able to search and view all of the bookmarks within the system, even without an account	H	3	✓
As a user, without an account, I do not want to be able to edit the bookmarks so that I don't interfere with company employees' work.	I want there to not be a clickable edit button for each bookmark	H	3	✓
As an admin, I want to be able to approve new accounts so that I can ensure only authorised users have full read write access to the bookmarks.	When a user signs up see their sign up request in a table including all their data Two options; approve, deny Once enabled ensure they have access to their account	H	3	✓
As an admin, I want to be able to disable accounts so that people who have for example left the company no longer	Want to be able to see a list of all the users, then disable an account, meaning the user can no	O	3	✓

have access.	longer log in using that account			
<b>Tagging System</b>				
As a user, I want to create tags so that bookmarks are categorised which makes them easier to search.	View a short textbox to set name	M	3	✓
As a user, I want to edit tags so that any errors can be corrected if they occur.	A button to click allowing me to edit An error if I attempt to change the name to a pre-existing tag	M	5	Not implemented because tags can be quickly added and deleted. Editing would not be as important.
As a user, I want to be able to delete tags so that tags without use can be removed.	When I click on a tag, I can delete it, meaning it is not associated with the tag anymore	O	3	✓
As a user, I want to easily give a reason to delete a tag so that the admin is aware of the change and why.	When a tag is deleted, a reason is given as to why the tag has been deleted, and this is stored in a 'change log' for an admin to view later	Ø	-3	Not implemented because when reporting a bookmark,
<b>Rating System</b>				
As a user, I want to be able to rate each bookmark out of 5, so that other users know the utility of that bookmark	Be presented with a rating scale, make my choice and view the updated rating on the bookmark.	M	3	✓
<b>Would be nice</b>				
As a user, I want to create a system with a playlist of favourites, so that I can quickly access these bookmarks	Have a list of favourites and also be able to unfavourite bookmarks	O	5	✓ <b>favourites page, not playlist</b>
As a user, I want to see a recently viewed list so I can go	Have a list of recently viewed bookmarks and to	Ø	3	Not implemented

back to the bookmark later	be able to visit those bookmarks from this list			as bookmarks can be easily searched
----------------------------	---	--	--	-------------------------------------

## 2.2 Additions

<b>Comment system</b>	As a user, I can add and delete comments on bookmarks so I can find out what the bookmark is for from other users	There is a comment box that can input description and be posted on the bookmark page

## 2.3 Index

### 2.3.1 Client Priority

H	High	Bare minimum
M	Medium	Minimum
O	Optional	If time allows

### 2.3.2 Estimated Effort

Effort ranked on the fibonacci scale (1, 2, 3, 5, 8, 13), 1 being the easiest in terms of effort, and 13 being the hardest.

## Part 3: Stories planned to be tackled in each iteration

Tasks planned to be tackled, listed in the general order of implementation.

### 3.1 Iteration 1

Login page	user can log in with the correct username and password stored in the database
Logout	user that is signed in can log out
View bookmarks	when clicking on the link to the bookmark, it takes the user to a page with additional information
Tags	user can add and remove tags in a bookmarks page, no duplicates can be added
Edit	user can edit name, description and resource

Search system	implementing basic search functionality
Comments	can add comments

[illegible]

Iteration 1	Overall systems	Total commits	Sprint weeks	target commits	actual commits	Completed tasks
	Database management	8	0	110	110	0
	Front-end web design	8	1	100	107	0
	Login system	12	2	90	107	0
	Bookmark system	14	3	80	101	0
	Search system	15	4	70	101	0
	Tag system	3	5	60	81	0
	Favourites system	1	6	50	63	0
	Admin	1	7	40	60	1
	Cucumber testing	8	8	30	51	0
	Report system	4	9	20	34	1
			10	10	3	2
			11	0	-12	2

Iteration 1 Burndown Chart



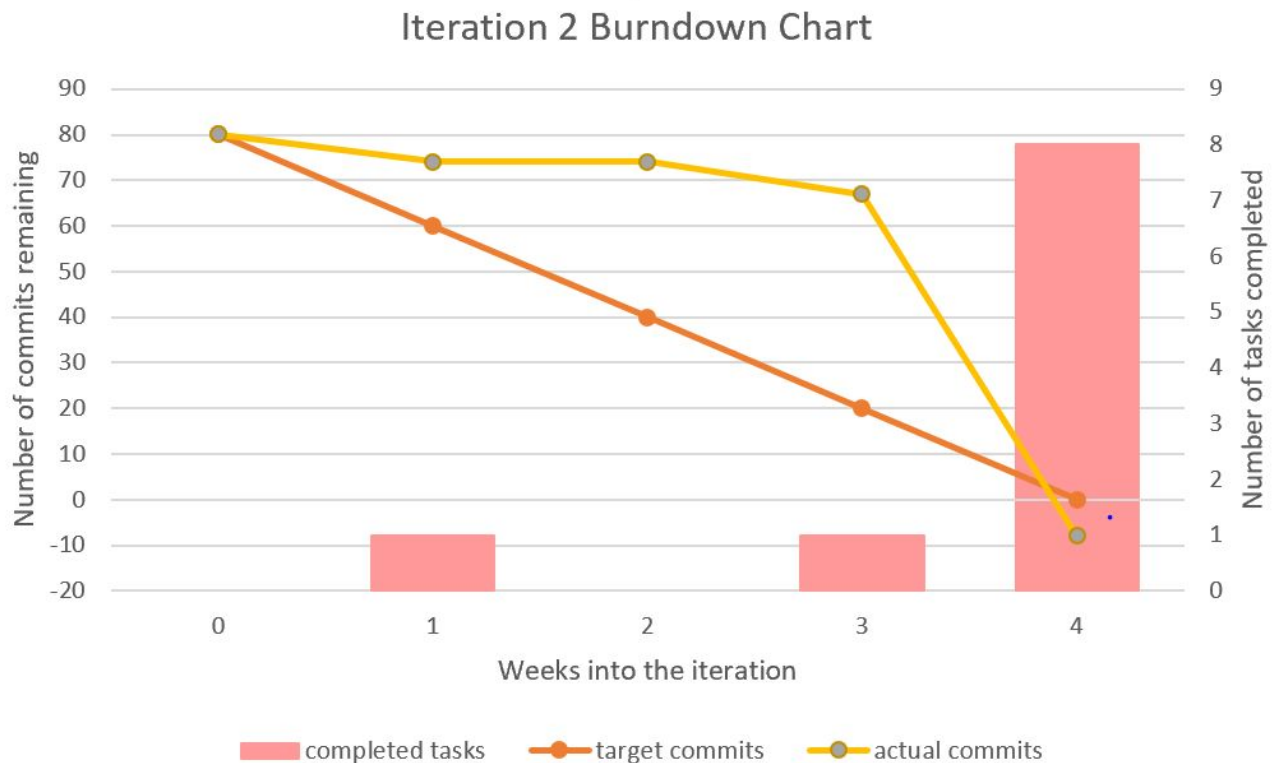
## 4.2 Iteration 2

	Tasks	Number of commits per week			
		28/04/2020	05/05/2020	12/05/2020	19/05/2020
Iteration 2	Front-end web design: nav bar	1	0	0	2
	Organising project file	0	0	0	0
	Editing system: finishing adjustments	5	0	0	0
	Tags: case sensitive and empty space error	0	0	0	2
	Favourites: finishing adjustments	0	0	0	2
	Favourites: testing	0	0	1	0
	Bookmarks: finishing adjustments	0	0	4	4
	Search system: search by tags	0	0	0	1
	Rating: adding ratings	0	0	0	9
	Rating: interface	0	0	0	3
	Admin page: account creation and deletion	0	0	2	6
	Admin page: account approval	0	0	0	7
	Admin page: view and delete reports	0	0	0	11
	Comments: delete comment & final adjustments	0	0	0	8
	Commenting the code	0	0	0	6
	Cucumber and rake	0	0	0	11

Iteration 2	Overall systems	Total commits
	Editing system	5
	Tags system	2
	Favourites system	3
	Bookmarks system	8
	Search system	1
	Rating system	12
	Admin system	26
	Comment system	14
	Cucumber and rake	11

Sprint weeks	target commits	actual commits	completed tasks
0	80	80	0
1	60	74	1
2	40	74	0
3	20	67	1
4	0	-5	8





## Part 5: Testing and coverage

Testing is split across our minitests and cucumber acceptance testing. Minitests located in "src/tests" are our unit tests designed to test the functions of every object found within "src/models". These unit tests also run integration tests as our models are linked to the database. This is why a separate database.db file exists in "src/tests" to act as a copy of our production database in "/src". This ensures the core features of the application run as expected. Acceptance tests with cucumber ensure the individual web pages load and execute correctly as the user sees them. Final user testing produced no noticeable errors.

### 5.1 Starting the minitests

```
"cd project/src/tests"  
"ruby alltest.rb "
```

### 5.2 Starting the Acceptance tests

```
"cd project/src"  
"rake feature"
```

