THE UNIVERSITY OF SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: SOFT2412 Agile Software Development Practices

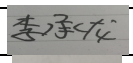
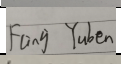
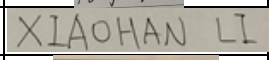
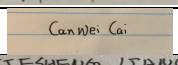
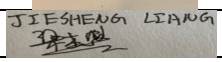
Assignment name: Agile Software Development with Scrum and Agile Tools - Sprint 1

Tutorial time: Thu 6p.m. - 8p.m. R18G3 Tutor name: Muhit Saleh Anik

DECLARATION
We the undersigned declare that we have read and understood the *University of Sydney Student Plagiarism: Coursework Policy and Procedure*, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that understand that failure to comply with the *Student Plagiarism: Coursework Policy and Procedure* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Project team members				
Student name	**Student ID**	**Participated**	**Agree to share**	**Signature**
1. Chenglong Li	**490035344**	Yes	Yes	
2. Yuben Fang	**480480835**	Yes	Yes	
3. Xiaohan Li	**470011746**	Yes	Yes	
4. Canwei Cai	**490032435**	Yes	Yes	
5. Jiesheng Liang	**480342832**	Yes	Yes	
6.		Yes / No	Yes / No	
7.		Yes / No	Yes / No	
8.		Yes / No	Yes / No	
9.		Yes / No	Yes / No	
10.		Yes / No	Yes / No	

Level 2, SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A

1.Scrum Development

1.1 Project Team

The role is determined in the meeting before Sprint 1 starts:
Product Owner: Jackson (Canwei) Cai
Scrum Master: Jacky (Xiaohan) Li
Core Team members: Jerry (Chenglong) Li, Lucius (Yuben) Fang, Jackson (Jiesheng) Liang

1.2 Sprint goal

In this Sprint, we aim to let the user successfully sign up and log in into their interface, they can put the items into the shopping cart and check out with the card. We are going to finish all the tasks including 6 User stories and divide them into 28 tasks in the sprint backlog. (This can refer to 3. Application development)

1.3 Tasks Board

1.3.1 Product backlog

We have 13 User Stories for the whole project. In this Sprint. There are 13 Story points in total.
1.As a customer, I want to set up an account so that I can buy something in the vending machine easily
2.As a registered customer, I want to Login my account using my username and password So that I can purchase from the Vending Machine.
3.As a registered customer, I want to put multiple items into a shopping cart so that I can pay for all the items in one transaction.
4.As an anonymous customer, I want to put multiple items into a shopping cart so that I can pay all the items in one transaction.
5. As a registered customer, I want to pay the transaction by card so that it is more convenient than cash.
6. As an anonymous customer, I want to pay the transaction by card so that it is more convenient than cash.
7.As a registered customer, I want to pay the transaction with cash so that I can pay without a credit card.
8. As an anonymous customer, I want to pay the transaction with cash so that I can pay without a credit card.
9.As a registered customer or anonymous customer, I want to pay the transaction with cash so that I can pay without a credit card.
10.As a seller, I want to modify items details and modify items so that I can add some goods into the vending machine.
11.As a Cashier, I want to modify the quantity of notes and coins in the machine so that I can.
12.As an Owner, I want all the ability of both Seller and Cashier (include the report) so that I have the highest right.
13.After 2 minutes, the system will close automatically with transaction uploaded

1.3.2 Sprint backlog

We choose 6 User Stories for this Sprint and divide each story into several tasks (see evidence).
Two user stories are completed in total:

User story 1: As a customer, I want to set up an account so that I can buy something in the vending machine easily. We divide this user story into 3 tasks.
1.Customer can create username and password
2.If the username has existed in the system, the sign-up operation failed, and customer will go back to the homepage
3.If the username is not in the system, the sign-up operation is successful, and the customer will go to the transaction part.

User story 2: As a registered customer, I want to Login my account using my username and password So that I can purchase from the Vending Machine.
We divide this user story into 3 tasks.
1.Customer can enter username and password
2.If the username has existed in the system and the password is correct, the login operation is successful, and the customer will go to the transaction part.
3.If the username does not in the system or the password is wrong, the login operation is not successful, and the customer will go back to the homepage

Remain four user stories are partially completed:
User story 3: As a registered customer, I want to put multiple items into a shopping cart so that I can pay for all the items in one transaction.
1.user can put products into a shopping cart unless the product is out of stock or the amount of products user input exceeds the inventory.
2.user can see all products in shopping cart and total price he needs to pay
3.a new transaction will be generated after login to record the detail of user's operation
4.the transaction will be automatically cancelled after 2 min (not completed yet)
5.5 items bought last time will appear in the interface (not completed yet)
6. Method used for cancelling the transaction (not completed yet)
7. Both failed and successful transaction will be uploaded to the database separately. (not completed yet)

User story 4: As an anonymous customer, I want to put multiple items into a shopping cart so that I can pay for all the items in one transaction.
1.user can put products into a shopping cart unless the product is out of stock or the amount of products user input exceeds the inventory.
2.user can see all products in shopping cart and total price he needs to pay
3.a new transaction will be generated after login to record the detail of user's operation
4.the transaction will be automatically cancelled after 2 min (not completed yet)
5.5 items bought last time by the previous anonymous user will appear in the interface (not completed yet)
6. Method used for cancelling the transaction (not completed yet)
7. Both failed and successful transaction will be uploaded to the database separately. (not completed yet)

User story 5: As a registered customer, I want to pay the transaction by card so that it is more convenient than cash.
User story 6: As an anonymous customer, I want to pay the transaction by card so that it is more convenient than cash.

Since the previous two stories have a lot in common, only anonymous users cannot save his/her card. We divide this into 8 tasks.
1.System should provide an interface to guide users to pay for items with a card that should include:
1. Provide your card details. 2. Cancel the transaction
2.If the user chooses '1' System should ask the user to input the cardholder's name and credit card number.
3.System should check if the card number matches the "credit_cards.json", System will output the error message and go back to let the user choose whether to input again.
4.The card number should appear as *.
5.The system can store card information as user's wish.
6.If the user chooses '2' System should check if there is a card saved in the system. If no, System will output "There is no match card!" and go back to the previous page.
7.The user can cancel that whole transaction in any time and the system will go back to the main page.
8.The system should go back to the main page after the transaction is done.
Only task 4 has not been implemented yet.

What's more, the burn-down chart is generated at the end of this sprint (see evidence).

1.4 Scrum Events Artefacts

1.4.1Sprint Planning

We planned to finish the login and signup functionality with the user object created. Then, the interface of choosing products and store them into a shopping cart. After that, we focused on two kinds of payment with Jason file read. Finally, we construct our database and connect the database to the backend.

1.4.2 Daily Scrum

As scrum defines, there should be a daily 15-minute standing up meeting. However, due to the particularity of the project, we decide to hold four meetings (see evidence) a week and extend the length of the meeting. For daily communication, we use WeChat (see evidence) to report the daily issues and progress.
In the first meeting, we define the construction of the database and start to design the interface of the product.
In the second meeting, we finish the interface parts and start to develop the allocated functionalities which are not connected to the database such as "sign up/log in", "check card valid", "add items to cart", etc.
In the third meeting, most of the functionalities are completed though some functionalities such as "record last 5 items" and "convert password and card number to *" are deferred to the next sprint since we need more time to complete the database and related functions.
In the last meeting, the database is successfully constructed, and we need to then complete the test case to reach the requirements of coverage and integrate our works together.

1.4.3 Challenge

1. The point of each user story is not that easy to judge. After a group meeting, we communicate with each other and set up the mark for each user story.
2. One member designed a database. However, he does not help other members to connect the database and cannot help other members to interact with the interface. So,

We must create a new database that everyone can use that database to interact with the interface. Finally, we overcome that and demonstrate the program with the client.
3.

1.4.4 Sprint review

In sprint 1, we held 4 meetings in total which made sure we have enough communication. However, we didn't have enough experience for managing the jira project, this led to many incorrect operations on converting our plan to tasks, sprints and Kanban board. We didn't assign appropriate story points to user stories and we thought only tasks needed to be put into sprint, this makes our burndown chart for the story point unavailable, however the burndown chart for issue count can be produced properly. In addition, the workload is too much, the expected efficiency is much higher than what we can achieve.

1.4.5 Sprint retrospective

For the next sprint, we are supposed to make some adjustments to the content of the user story which should be more specific to differentiate the task. In addition, we need to learn more about the settings of Jira. As for user stories, the story point should be estimated appropriately in advance. In this case, a story burndown chart can be created to show more information about our performance in the whole project process. The tasks can also be distributed proper for each sprint. Additionally, the sprint board needs to be remembered to update every day, which ensures a correct issue burndown chart can be generated.

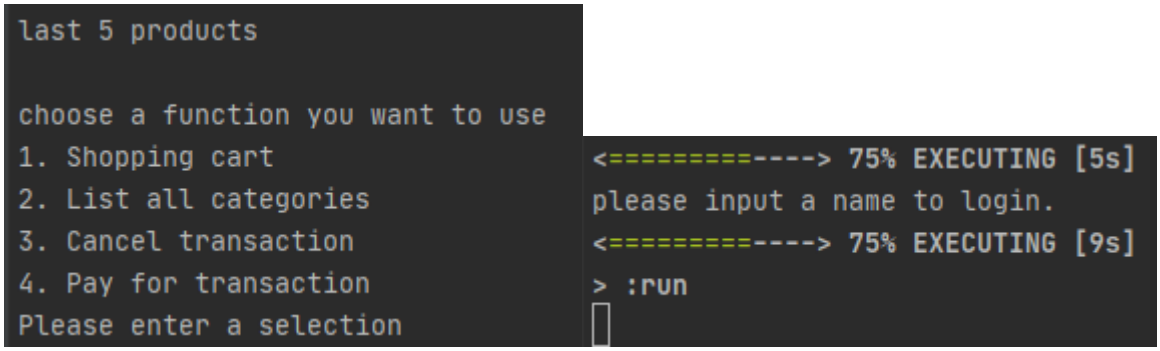
2. Agile development tools

We use github, gradle, junit and jenkins for this project as our tools to develop the product. In the github, members will create branches when they need to implement new features on the product. For gradle, we add dependencies so that we can use junit to test codes and generate jacoco test reports and also allow us to use the sql database. Using jenkins, we can apply our continuous integration on our project, it provides functionality like automated build and test. Every time group members push their code to the git repository; they can see their build and test result on jenkins server which is maintained by one of our members.

3. Application development

3.1 User interface

Simple command line written in Java is used as our main interface since it is simple and able to complete all functions. The user can choose functions to run with numbers which the system offers and he/she can also type in command from keyboards when the system needs. The user can simply terminate the system by "ctrl + c".




```
Hi jackson nice to meet you :)
last 5 products

choose a function you want to use
1. Shopping cart
2. List all categories
3. Cancel transaction
4. Pay for transaction
Please enter a selection
```

3.2 Functionality

We developed 4 functionalities in sprint 1: login, sign up, add items into cart and card payment.

3.2.1 Login

Customers can enter their username and password to go to their specific transaction interface.

3.2.2 Sign Up

After customer input and create their username and password, the system will check whether the username is in the database or not. If the username has been registered in the database, the system will output "Sorry, there is the same user name in the system, please enter another username to create an account" and create unsuccessfully. If the username is not in the database, the customer can create the account and automatically go to the special transaction interface.

3.2.3 Add Items into Cart

Customers can check products with their information (name, quantity in stock and price) in four categories (Drinks,Chocolates,Chips,Candies) then add them into the shopping cart so that they can pay for them later. The shopping cart can save all selected items with item name and how many this kind of item is added into cart, also it will show the total price of all items in the shopping cart.

3.2.4 Card Payment

Both registered customers and anonymous customers can use card payment to checkout their shopping cart. They should provide valid card details including card holder and card number, only the registered customers can save the card details after a successful transaction. Then the transaction details would be recorded in the database. If the transaction is cancelled, cancelled detail should be recorded in the database and the system should go back to the home page.

3.2.5 Database

The class ‘ConnectSQLite’ is used for creating a local sqlite database, generating the schema of each table and initialing some data before the whole program runs. The database is created in a local

repository file called ‘sample.db’. After that, we can connect to the database from the backend, obtain and send data from/to the database.

Item		
Item_name	PK	String
Item_category	not NULL	String(Drink or Chocolate or Chip or Candy)
Item_price	not NULL	Double
item_quantity_remain	not NULL	Int
Item_quantity_total_sold	not NULL	Int

User		
User_name	PK	String("anonymous" if anonymous user)
User_password	defalut NULL	String("NULL" if anonymous user)
User_last_5_items	defalut NULL	StringList
Card_holder_name	defalut NULL	String
Card_number	defalut NULL	String
User_type	not NULL	String(Customer,Seller,Cashier,Owner,Anonymous)

Change		
Change_id	PK	Int
Change_type	not NULL	Double(100 or 50 or 20 or 10 or 5 or 2 or 1 or 0.5 or 0.2 or 0.1 or 0.05)
Change_quantity	not NULL	Int

Transaction		
Transaction_id	PK	Int
Transaction_date	not NULL	Date
Transaction_time	not NULL	Time
Transaction_amount	not NULL	Double
Transaciton_items	not NULL	String(e.g. 1xcola,2xsneaker)
Transaction_change	not NULL	Double
User_name	FK	String
Transaction_method	not NULL	String(Cash or Card)

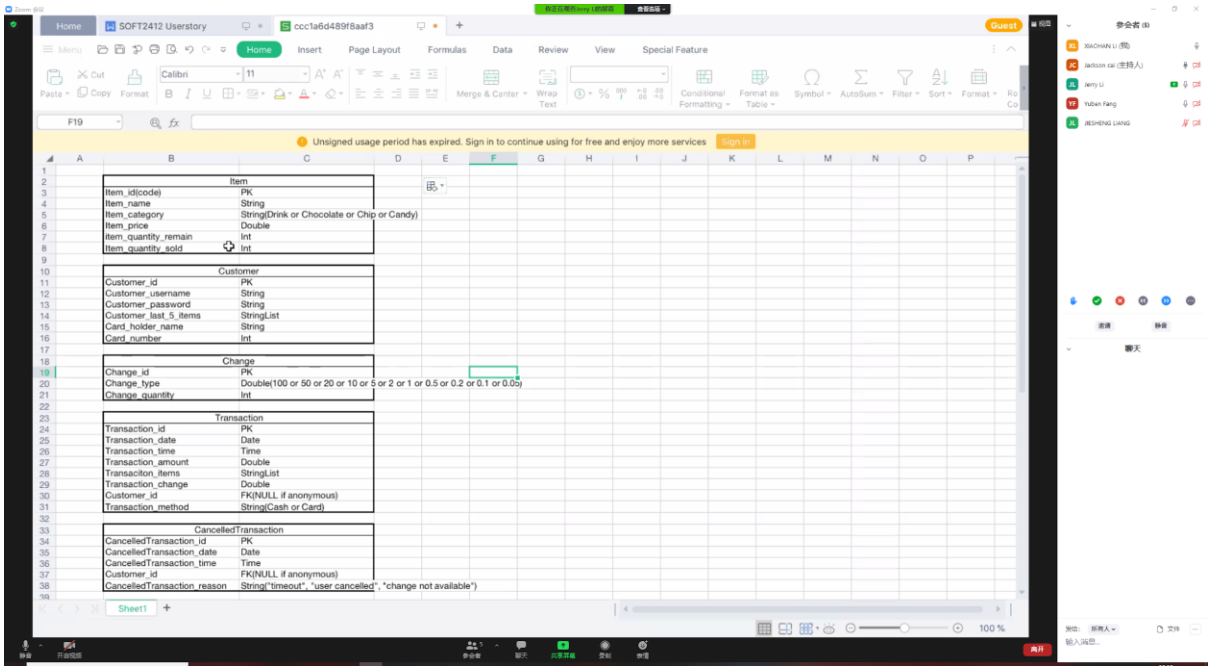
CancelledTransaction		
CancelledTransaction_id	PK	Int
CancelledTransaction_date	not NULL	Date
CancelledTransaction_time	not NULL	Time
User_name	FK	String
CancelledTransaction_reason	not NULL	String("timeout", "user cancelled", "change not available")

3.4 Contribution

Jackson (Canwei) Cai - User story design, Sprint planning, User story 1&2 implementation, Sprint-1-report
Jacky (Xiaohan) Li - Database design, Scrum event records, User story 5&6 implementation, Sprint-1-report
Jerry (Chenglong) Li - Agile tools management, User story 3&4 implementation, Sprint-1-report
Lucius (Yuben) Fang - Agile tools management, User story 3&4 implementation, Sprint-1-report
Jackson (Jiesheng) Liang - Database construction, Sprint-1-report

4. Evidence

Scrum development:
Scrum events artefacts:
Daily scrum



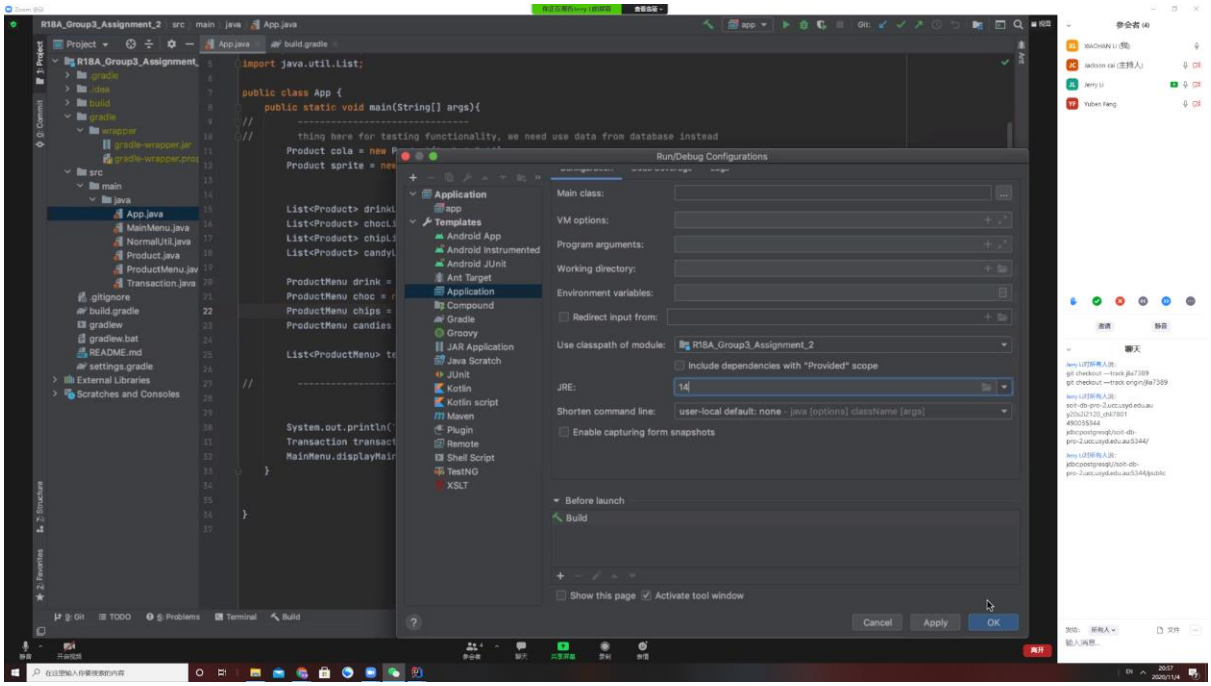
Item	
Item_id(code)	PK
Item_name	String
Item_category	String(Drink or Chocolate or Chip or Candy)
Item_price	Double
item_quantity_remain	Int
Item_quantity_sold	Int

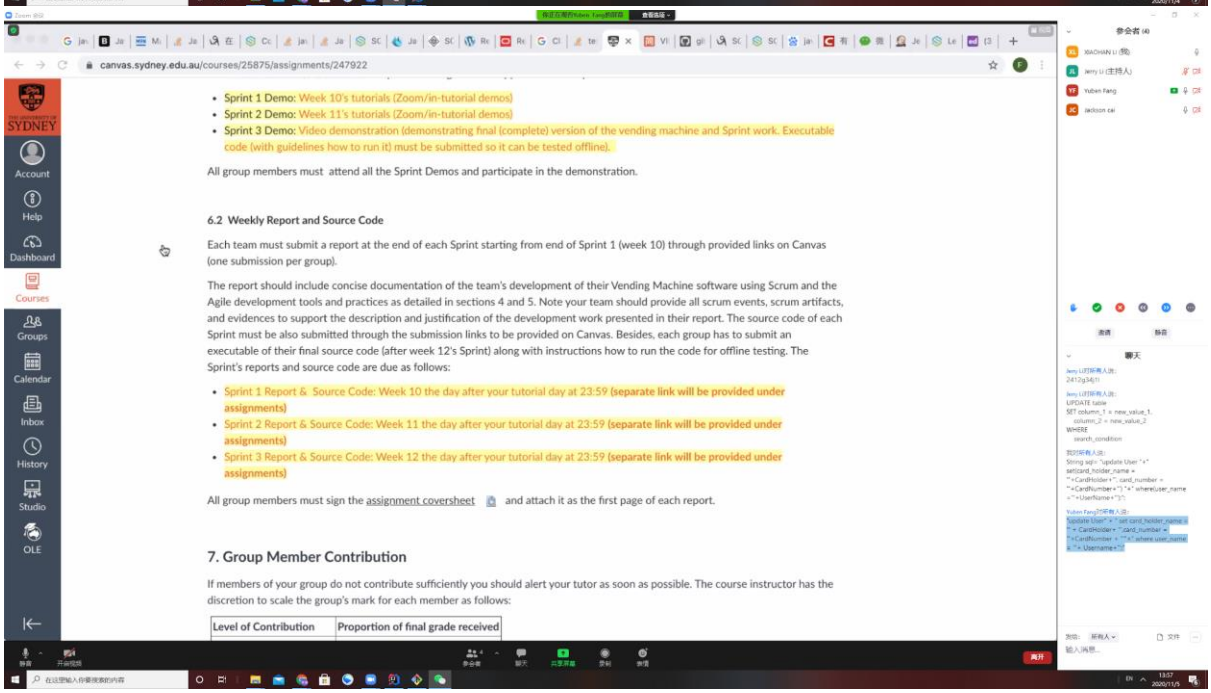
Customer	
Customer_id	PK
Customer_username	String
Customer_password	String
Customer_last_5_items	StringList
Card_holder_name	String
Card_number	Int

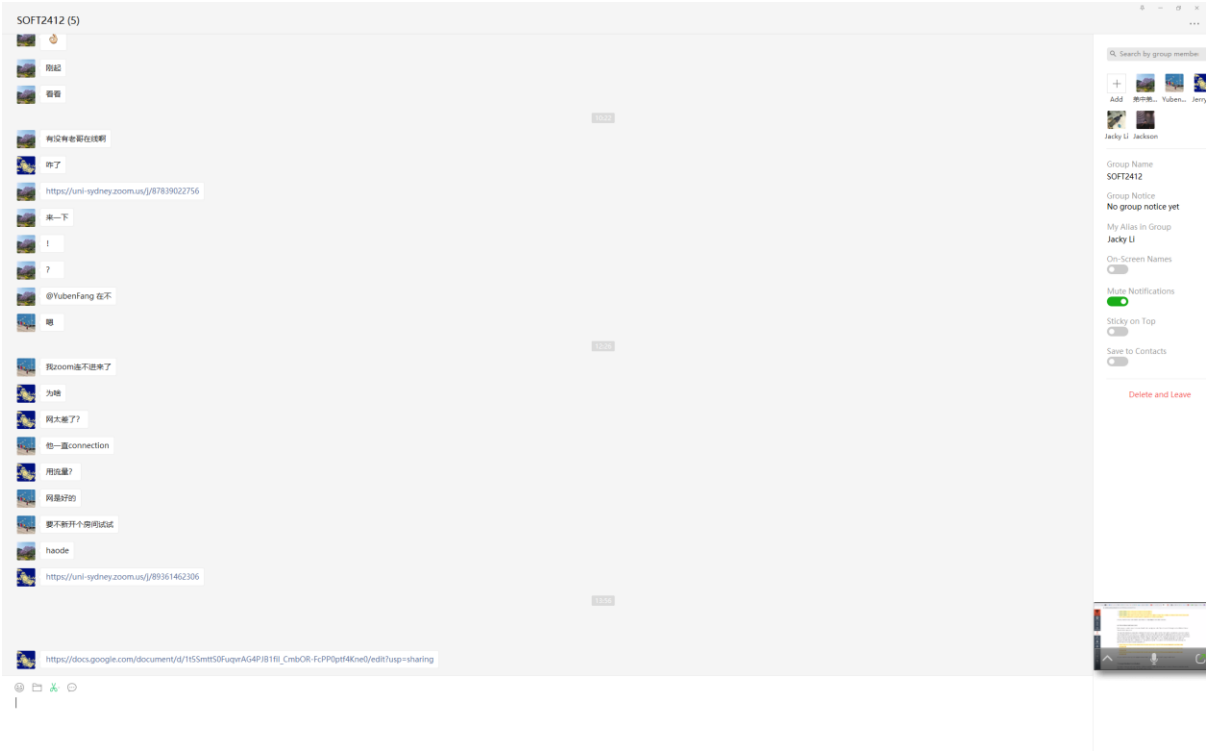
Change	
Change_id	PK
Change_type	Double(100 or 50 or 20 or 10 or 5 or 2 or 1 or 0.5 or 0.2 or 0.1 or 0.05)
Change_quantity	Int

Transaction	
Transaction_id	PK
Transaction_date	Date
Transaction_time	Time
Transaction_amount	Double
Transaciton_items	StringList
Transaction_change	Double
Customer_id	FK(NULL if anonymous)
Transaction_method	String(Cash or Card)

CancelledTransaction	
CancelledTransaction_id	PK
CancelledTransaction_date	Date
CancelledTransaction_time	Time
Customer_id	FK(NULL if anonymous)
CancelledTransaction_reason	String("timeout", "user cancelled", "change not available")

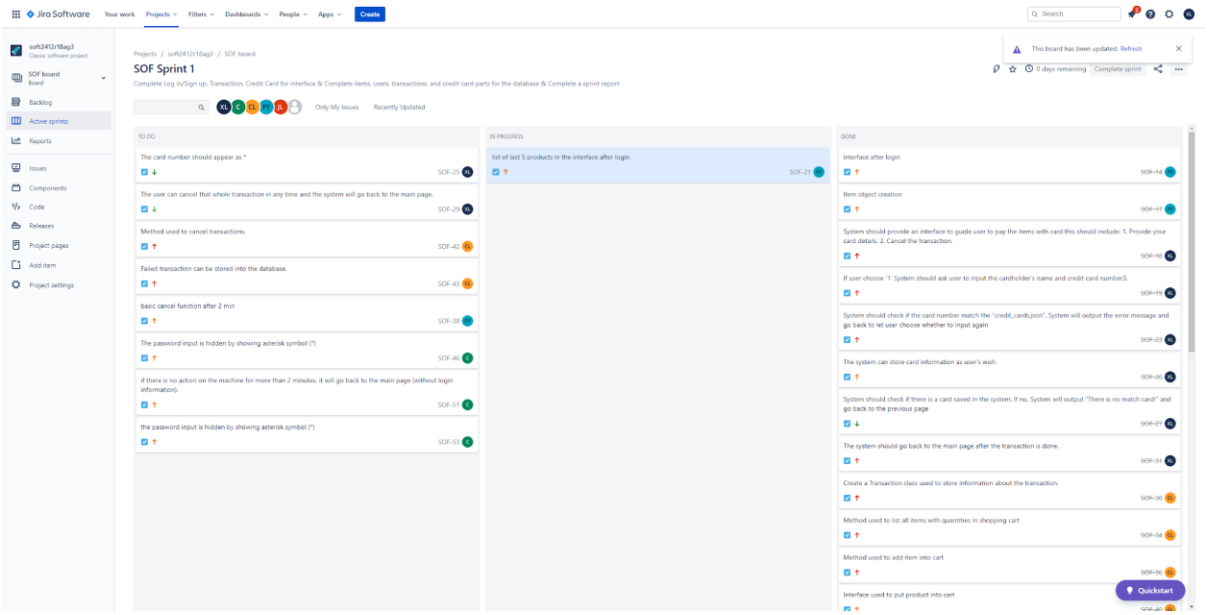

Run/Debug Configurations
Application
Templates
Main class:
VM options:
Program arguments:
Working directory:
Environment variables:
Redirect input from:
Use classpath of module:
R18A_Group3_Assignment_2
Include dependencies with "Provided" scope
JRE:
14
Shorten command line:
user-local default: none - java [options] className [args]
Enable capturing form snapshots
Before launch
Build
Show this page
Activate tool window
Cancel
Apply
OK

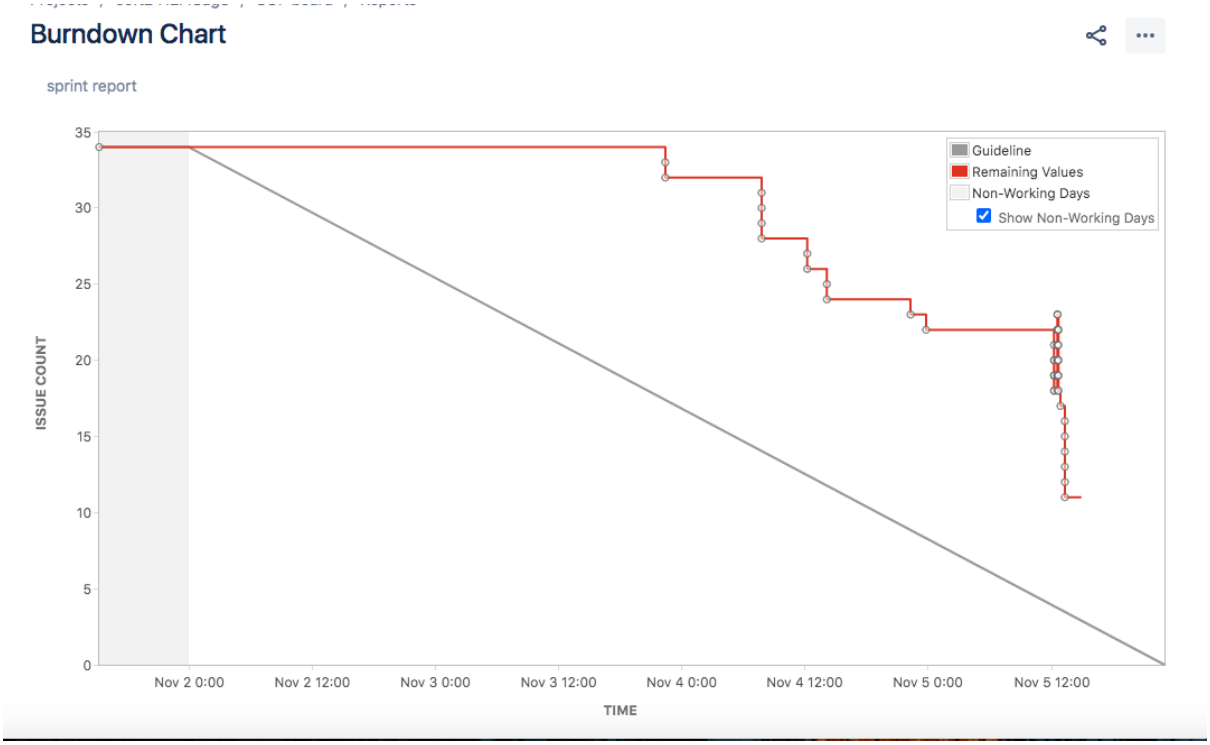

canvas.sydney.edu.au/courses/25875/assignments/247922
Sprint 1 Demo: Week 10's tutorials (Zoom/in-tutorial demos)
Sprint 2 Demo: Week 11's tutorials (Zoom/in-tutorial demos)
Sprint 3 Demo: Video demonstration (demonstrating final (complete) version of the vending machine and Sprint work. Executable code (with guidelines how to run it) must be submitted so it can be tested offline).
All group members must attend all the Sprint Demos and participate in the demonstration.
6.2 Weekly Report and Source Code
Each team must submit a report at the end of each Sprint starting from end of Sprint 1 (week 10) through provided links on Canvas (one submission per group).
The report should include concise documentation of the team's development of their Vending Machine software using Scrum and the Agile development tools and practices as detailed in sections 4 and 5. Note your team should provide all scrum events, scrum artifacts, and evidences to support the description and justification of the development work presented in their report. The source code of each Sprint must be also submitted through the submission links to be provided on Canvas. Besides, each group has to submit an executable of their final source code (after week 12's Sprint) along with instructions how to run the code for offline testing. The Sprint's reports and source code are due as follows:
Sprint 1 Report & Source Code: Week 10 the day after your tutorial day at 23:59 (separate link will be provided under assignments)
Sprint 2 Report & Source Code: Week 11 the day after your tutorial day at 23:59 (separate link will be provided under assignments)
Sprint 3 Report & Source Code: Week 12 the day after your tutorial day at 23:59 (separate link will be provided under assignments)
All group members must sign the assignment coversheet and attach it as the first page of each report.
7. Group Member Contribution
If members of your group do not contribute sufficiently you should alert your tutor as soon as possible. The course instructor has the discretion to scale the group's mark for each member as follows:
Level of Contribution
Proportion of final grade received



Scrum artefacts:

Sprint backlog





Jenkins jacoco test coverage:

name	instruction	branch	complexity	line	method	class
(default)	**M:** 858 **C:** 1293 60%	**M:** 86 **C:** 58 40%	**M:** 75 **C:** 111 60%	**M:** 244 **C:** 344 59%	**M:** 21 **C:** 84 80%	**M:** 2 **C:** 16 89%