Tool-support for modelling and analysis of data and control flow dependencies

19 October 2015

Project Supervisor: Maria Spichkova

Submitted by Group 100:

3381049 Gor;Simon Paul

3207568 Kim;Deok Yeon

3318707 Yahya;Zohoor

3405533 Deng;Yang

1 CONTENTS

2 Project Charter 4
2.1 Introduction 4
2.1.1 Purpose of project charter 4
2.2 Project objective 4
2.3 Overview 5
2.3.1 Objectives 5
2.3.2 High-level requirements .... **Error! Bookmark not defined.**
2.3.3 Executive milestones 6
2.4 Assumptions, Constraints and Risks 6
2.4.1 Assumption (2 to 3 points should be added) 6
2.4.2 Constraints 6
2.4.3 Risks 7
2.5 Project Organization 8
2.5.1 Roles and Responsibilities of project members 8
3 Minutes of weekly planning sessions and updated tasks 9
4 System Design Document (in progress) 27
4.1 GUI (graphical user interface) 27
4.2 Upload L0 29
4.3 Display L0 30
4.4 Check elementary service 30
4.5 Generate L1 (needs fixing) 31
4.6 Find strongly connected service 32
4.7 Generate L2 33
4.8 Remote computation 33
4.9 Check property 34
4.10 Instructions on how to use eclipse 36
4.11 Diagram of system with new names 41
4.11.1 Level L0 system 41
4.11.2 Level L1 system 41
4.11.3 Level L1 detection of strongly connected service 42
4.11.4 Level L2 system 42

4.11.5 Level L3 system ... 43
5 Issues and Risk Registers (In progress) ... 43
6 Test Cases (In Progress) ... 45
7 Peer Review (waiting for Yang’s peer review) ... 51
7.1 Simon’s peer review: ... 51
7.2 Zohoor’s peer review: ... 52
7.3 Deok’s peer review: ... 53
7.4 Yang’s Peer review : ... 54
8 Technical Solution Design Document (In progress) ... 56
8.1 Why we chose java over python ... 56
8.2 Why we chose Tarjan over One Way Catch Them young (OWCTY) ... 56
8.2.1 What is Tarjan algorithm? ... 56
8.3 Why we pick JSON as an input file ... 57
9 Learning Outcomes ... 58
9.1 Enabling Knowledge ... 58
9.2 Critical Analysis ... 58
9.3 Problem solving ... 59
9.4 Communication ... 59
9.5 Teamwork ... 60

2 PROJECT CHARTER

2.1 INTRODUCTION

2.1.1 Purpose of project charter

The project charter defines the scope, objectives, and overall approach for the work to be completed. It is a critical element for initiating, planning, executing, controlling, and assessing the project. It should be the single point of reference on the project for project goals and objectives, scope, organization, estimates, work plan. In addition, it serves as a contract between Maria Spichkova who will be acting as our supervisor and our team , stating what will be delivered according to the time constraints, risks, resources, and standards agreed upon for the project.

2.2 PROJECT OBJECTIVE

In this project, we will be working on creating an application which is based on the ideas on the publication that called ‘Towards logical architecture and formal analysis of dependencies between services’ by Maria Spichkova and Heinrich Schmidt. This project will allow the services decomposition and analyzing of dependency within a system.

Five stages of the system analysis

1. Structure
2. Elementary services
3. Strongly connected services
4. Efficient checking of properties
5. Remote computation (Cloud)

2.3 OVERVIEW

2.3.1 Objectives

The objectives of the project are listed as follows

- Enable analysis of data/control flow dependencies
- Making it error-free
- Users will be able to open and save the system.
- Decompose any designed system based on given reliability characteristics of sys tem's components.
- Trace interconnection between subsystems
- Identify strongly connected services
- User friendly and reliable by using intuitive GUI
- Provide user manual to aid users

2.3.2 Executive milestones

The table below lists the executive milestones of the project and their estimated completion timeframe.

Executive Milestone	Estimated Completion Timeframe
Development and implementation	
Analysis of data of each level	All the analysis of each level of .(14 days)
Decomposition into elementary service	2 weeks
Identify strongly connected services	2 weeks
Efficient checking of properties	2 weeks
Remote computation	2 weeks
Test tools functionality	1 week
Design GUI for tools	2 week

2.4 Assumptions, Constraints and Risks

2.4.1 Assumption (2 to 3 points should be added)

Meetings with supervisor

It is important that the project group meetings with supervisor is conducted weekly it is one of the ways to keep each other updated with work matters and a way of keeping track of individual and team work progress through a healthy discussion. Therefore, the whole team is constantly updated with how well the project goes and what things need to be worked on and if some modification is needed, the supervisor will be able to keep the group in the right track.

2.4.2 Constraints

Deadline

A time period of 12 weeks has been allocated to complete the project. Meeting the timeline is a significant aspect. After the milestone is set, it is best not to have any kinds of procrastination which will cause extra workload.

Understanding algorithms

Understanding algorithms in the article is a significant key to complete the project. Based on the article given, each member should be able to understand the algorithms.

2.4.3 Risks

Risks	Effect	Counter measure
1) Incorrect understanding of the requirements of the project goal or a flaw in the report	This may lead to major damage and delay to the project, if not discovered on time or goes undetected, the software will be developed in unintended way which may or may not be able to meet the ultimate aim or requirements of the project.	There should be through requirements analysis done and revise before the actual project is ready to begin. There must be constant checks carried out to make sure the needs and requirements are being adhered to and there aren't any mismatches. If any abnormalities are detected, it must be rectified.
2) Delay in software and report	It is critical to know that delay in the project on when and how the incident occurs, at certain time of ongoing process will be likely held up until appropriate action is established. It could be a minor at the earlier stage of the project, but it will be severe as time passes	Self-knowledge is important for every member of the team that individual works must be frequent backed up and distribution should be made to the others. Also, project manager should be able to check the allocated work to each member which will be an alert that process is on the right tract.
3) Communication	Lack of communication among the group members and the supervisor will cause the misunderstanding towards the project in a way of distributions in programming work and delivery of report is not done by individually, when it comes to merging each members work and the weekly meeting with supervisor should be conducted well.	Every group member is cautious about the notification from the supervisor or project manager. When unexpected incident happens, notify the group member in order to prevent the accident.

2.5 PROJECT ORGANIZATION

2.5.1 Roles and Responsibilities of project members

This section describes the key roles supporting the project.

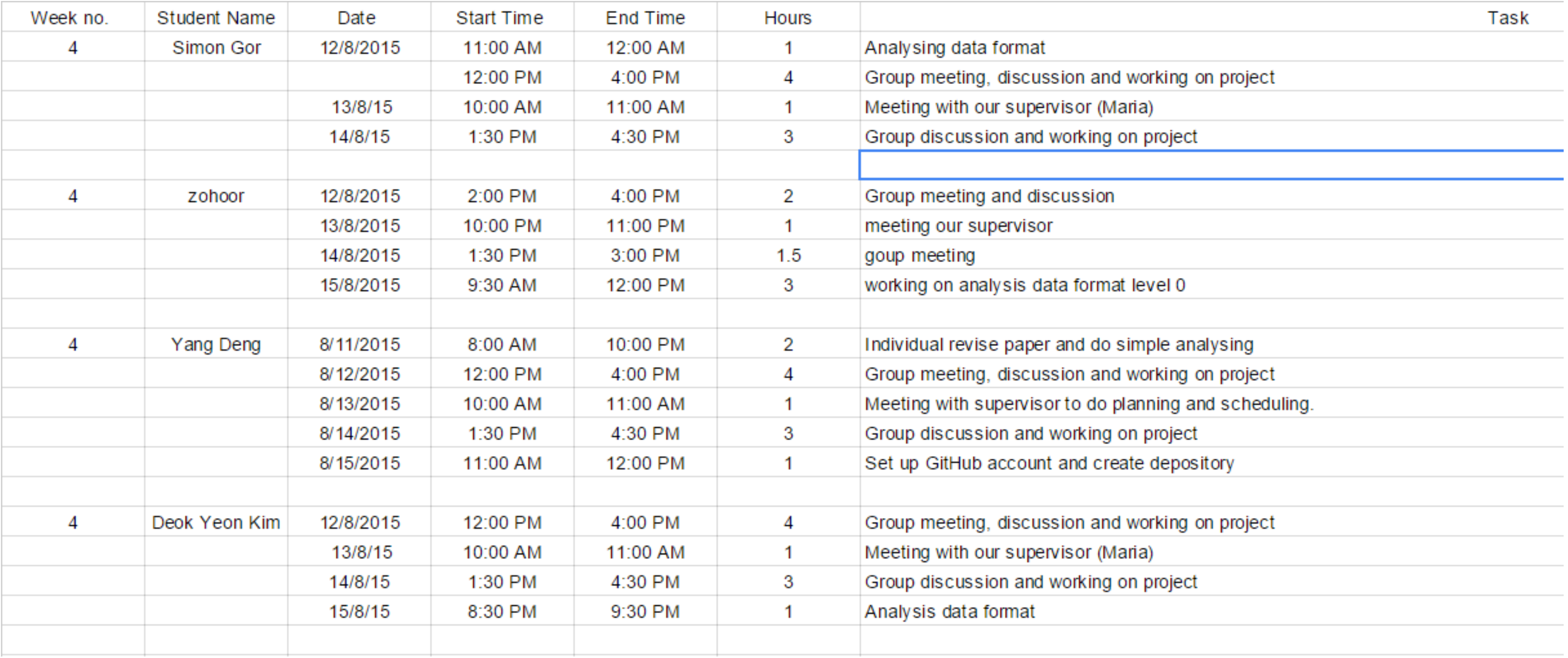
Name	Project role	Project responsibilities
Maria Spichkova	Project supervisor	Evaluates the project and encourage and assist the project group to develop standards of achievement.
Simon Gor	Project manager	Overall responsibility for successful initiating, planning, design, execution, monitoring, controlling and closure of the project. Also, recognize the risks at times
Zohoor Yahya	Developer, Tester, Report contributor	Translate software requirements into workable programming code and maintain and develop the programs as well as analyzing, reviewing the requirements and design specifications.
Yang Deng	Developer, Tester, Report contributor	Translate software requirements into workable programming code and maintain and develop the program as well as analyzing, reviewing the requirements and design specifications.
Deok Yeon Kim	Report writer, Tester	Testing is a fundamental aspect of the role so that the application can be analyzed in a proper way and communicates with developers to exchange the findings and results therefore building and repairing the report which will be provided to end-users.

3 MINUTES OF WEEKLY PLANNING SESSIONS AND UPDATED TASKS

Week	**4**
Completed tasks/features	-Analysis of data format: As a group we looked at different types of input data to use for our tool. We decided to use Json data format since there are many libraries available to assist us with the implementation process. -Further analysis of the paper: Upon deciding the data format, we thoroughly read through the paper and held a discussion session. During this discussion session we answered any queries/questions about the paper in order for us to understand it better. -Planning: This week we also planned on how to create the software. E.g. GUI, output design, functionality and features.
Changes to tasks/features	There were no changes to tasks or software features.
Current issues	-Contemplating whether our planning and design for the software was correct. -Some group members found it hard to understand the different ideas/concepts represented in the paper
Fixed issues	-There were no issues to fix this week.
Current tasks (started working on)	-After completing this week's tasks we didn't start on anything else.
Future tasks	-Create a L0 input file -Create software -Create GUI -Upload and read input file -Check elementary service -Generate L1 -Find strongly connected service -Generate L2 -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Project charter -System design documents -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

Individual contributions (incl. timesheets):

Name	Contribution/Summary
Simon Gor	This week I focused on understanding the paper provided by Maria. Once I thoroughly read through each concept, I represented my understanding in a document for each group member to read. These concepts will then be used for developing our software. Apart from reading the paper I also helped with deciding the input data type and plan our software.
Zohoor Yahya	This week Zohoor also focused understanding the concepts/ideas within the paper provided by Maria. Zohoor represented her understanding of the paper in a document to share and compare with each group member. Zohoor also helped decide on the input data type and plan the software.
Yang Deng	This week Yang attempted to understand the concepts/ideas represented in the paper. But due to his language barrier he found it difficult to read the formal representation throughout the paper. Although Yang couldn't represent his understanding in document, he helped with planning the software and deciding on the input data type.
Deok Kim	This week Deok also attempted to understand the concepts/ideas represented in the paper. He also found it difficult to understand due to his language barrier. Even through Deok couldn't represent his understanding in a document he helped the group plan our software and decide on the input data type.

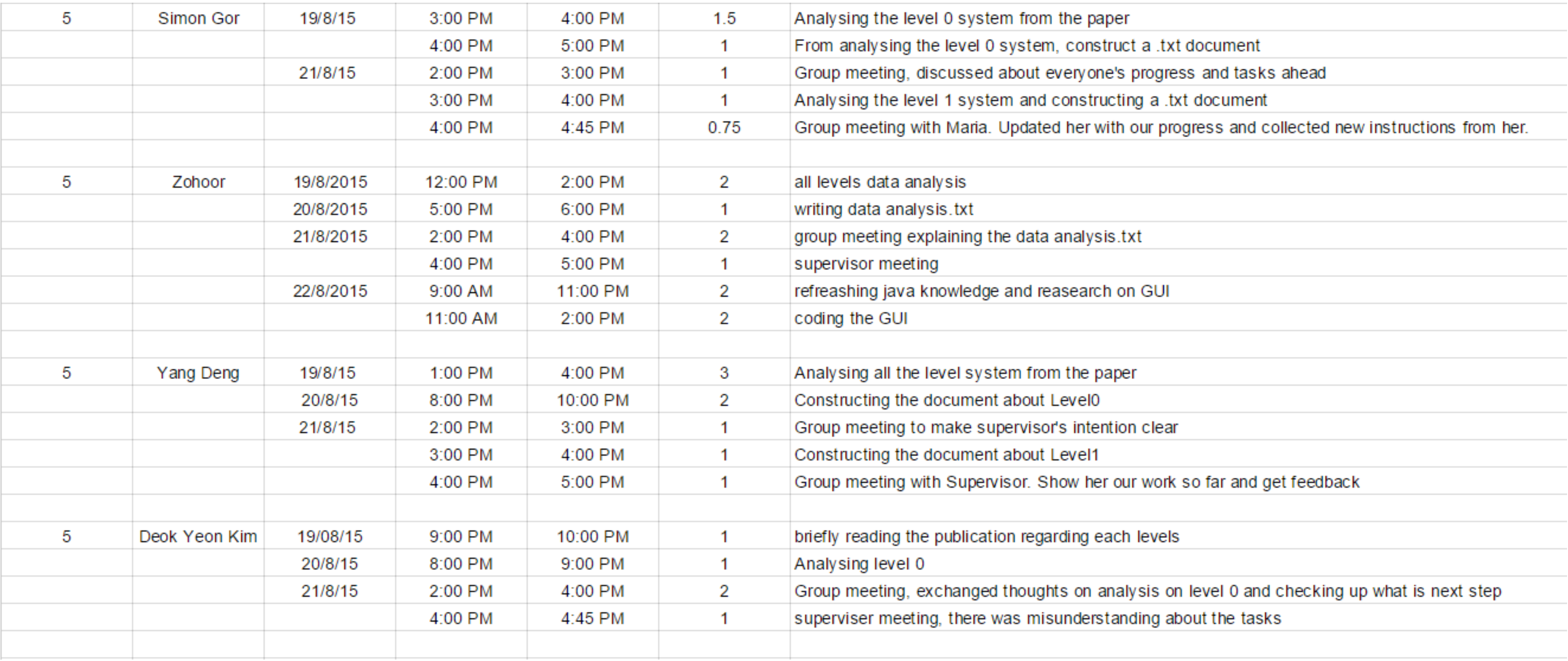


Week no.	Student Name	Date	Start Time	End Time	Hours	Task
4	Simon Gor	12/8/2015	11:00 AM	12:00 AM	1	Analysing data format
			12:00 PM	4:00 PM	4	Group meeting, discussion and working on project
		13/8/15	10:00 AM	11:00 AM	1	Meeting with our supervisor (Maria)
		14/8/15	1:30 PM	4:30 PM	3	Group discussion and working on project
4	zohoor	12/8/2015	2:00 PM	4:00 PM	2	Group meeting and discussion
		13/8/2015	10:00 PM	11:00 PM	1	meeting our supervisor
		14/8/2015	1:30 PM	3:00 PM	1.5	goup meeting
		15/8/2015	9:30 AM	12:00 PM	3	working on analysis data format level 0
4	Yang Deng	8/11/2015	8:00 AM	10:00 PM	2	Individual revise paper and do simple analysing
		8/12/2015	12:00 PM	4:00 PM	4	Group meeting, discussion and working on project
		8/13/2015	10:00 AM	11:00 AM	1	Meeting with supervisor to do planning and scheduling.
		8/14/2015	1:30 PM	4:30 PM	3	Group discussion and working on project
		8/15/2015	11:00 AM	12:00 PM	1	Set up GitHub account and create depository
4	Deok Yeon Kim	12/8/2015	12:00 PM	4:00 PM	4	Group meeting, discussion and working on project
		13/8/15	10:00 AM	11:00 AM	1	Meeting with our supervisor (Maria)
		14/8/15	1:30 PM	4:30 PM	3	Group discussion and working on project
		15/8/15	8:30 PM	9:30 PM	1	Analysis data format

Week	**5**
Completed tasks/features	-Create a L0 input file: During our group meeting this week we worked together on the L0 input file and completed it. -Create a L1 input file: We also finished creating the L1 input file which wasn't needed in the end.
Changes to tasks/features	There were no changes to tasks or software features.
Current issues	-After our meeting this week with Maria, our group found out that we were on the wrong track and misunderstood the instructions. Due to this we fell behind in our work and had to catch up. -Haven't started creating our software when we were supposed to complete the features: Upload and read input file, check elementary service and generate L1. -Haven't started documentation since we misunderstood Maria's instructions.
Fixed issues	-Contemplating whether our planning and design for the software was correct. During our meeting this week, Maria gave us a hand with our software plan to make sure we are on the right track. -Some group members found it hard to understand the different ideas/concepts represented in the paper. Yang and Deok finally understood the ideas/concepts through the help of Zohoor and myself.
Current tasks (started working on)	-Create software -Create GUI
Future tasks	-Create software -Upload and read input file -Check elementary service -Generate L1 -Find strongly connected service -Generate L2 -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Project charter -System design documents -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

Individual contributions (incl. timesheets):

Name	Contribution/Summary
Simon Gor	During this week I worked with my group members to complete the L0 input text file. Then at home I completed the L1 input file and compared it with my group members.
Zohoor Yahya	This week Zohoor worked with the group to complete the L0 input text file. She also completed the L1 input file at home and compared it with my L1 file. During the weekend, Zohoor started working on our software and coded the GUI.
Yang Deng	In the early parts of the week, Yang did some further analysis on the paper in order to understand it better. Later on Yang also worked with the group and contributed to the L0 input text file. He did his own representation of the L0 input .txt file at home since he wasn’t confident of the structure and content. After he finished the L0 input file, he successfully completed the L1 input .txt file on his own.
Deok Kim	This week Deok continued his analysis of the paper (early in the week) in order to understand it better and contributed to the L0 input text file.

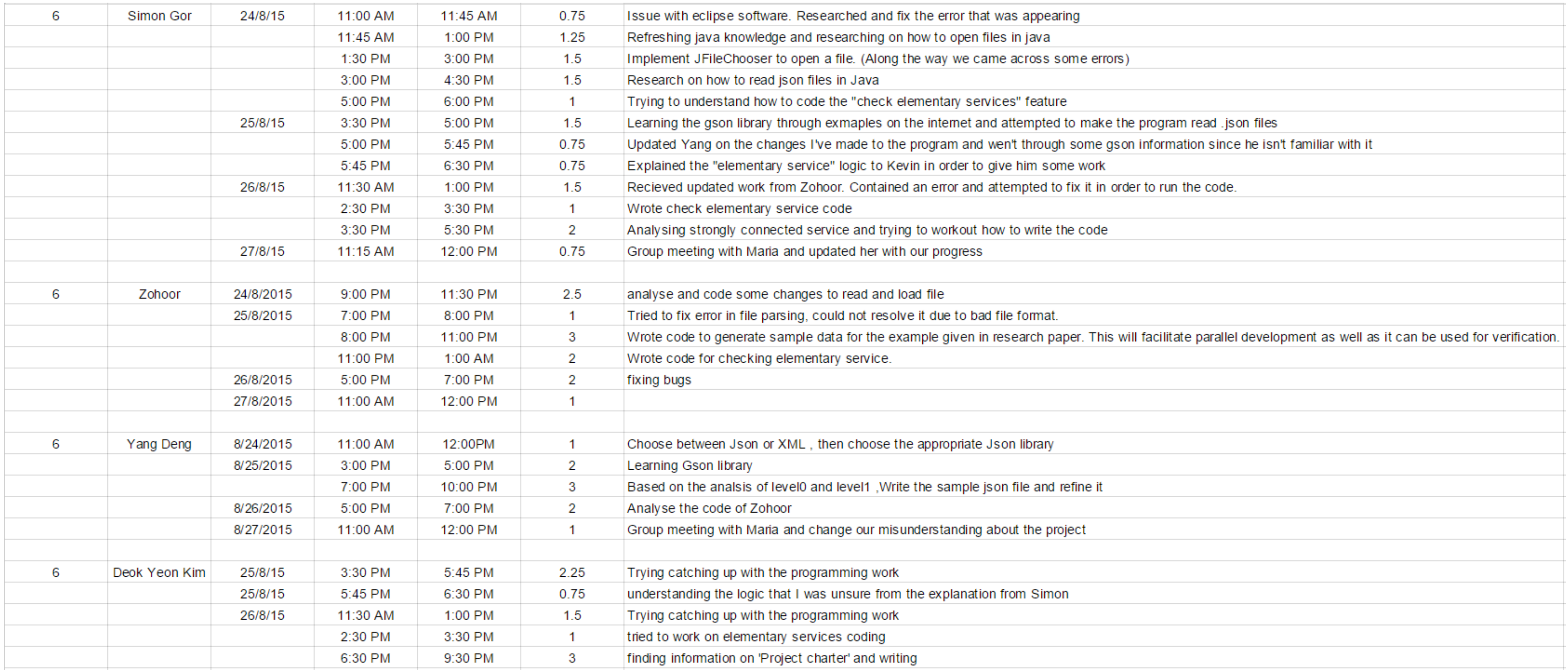


5	Simon Gor	19/8/15	3:00 PM	4:00 PM	1.5	Analysing the level 0 system from the paper
			4:00 PM	5:00 PM	1	From analysing the level 0 system, construct a .txt document
		21/8/15	2:00 PM	3:00 PM	1	Group meeting, discussed about everyone's progress and tasks ahead
			3:00 PM	4:00 PM	1	Analysing the level 1 system and constructing a .txt document
			4:00 PM	4:45 PM	0.75	Group meeting with Maria. Updated her with our progress and collected new instructions from her.
5	Zohoor	19/8/2015	12:00 PM	2:00 PM	2	all levels data analysis
		20/8/2015	5:00 PM	6:00 PM	1	writing data analysis.txt
		21/8/2015	2:00 PM	4:00 PM	2	group meeting explaining the data analysis.txt
			4:00 PM	5:00 PM	1	supervisor meeting
		22/8/2015	9:00 AM	11:00 PM	2	refreashing java knowledge and reasearch on GUI
			11:00 AM	2:00 PM	2	coding the GUI
5	Yang Deng	19/8/15	1:00 PM	4:00 PM	3	Analysing all the level system from the paper
		20/8/15	8:00 PM	10:00 PM	2	Constructing the document about Level0
		21/8/15	2:00 PM	3:00 PM	1	Group meeting to make supervisor's intention clear
			3:00 PM	4:00 PM	1	Constructing the document about Level1
			4:00 PM	5:00 PM	1	Group meeting with Supervisor. Show her our work so far and get feedback
5	Deok Yeon Kim	19/08/15	9:00 PM	10:00 PM	1	briefly reading the publication regarding each levels
		20/8/15	8:00 PM	9:00 PM	1	Analysing level 0
		21/8/15	2:00 PM	4:00 PM	2	Group meeting, exchanged thoughts on analysis on level 0 and checking up what is next step
			4:00 PM	4:45 PM	1	superviser meeting, there was misunderstanding about the tasks

Week	**6**
Completed tasks/features	-Create GUI: We have successfully implemented a simple GUI that will allow users to use all of the intended functions (Generate L0 and L1 etc.). -Check elementary service: The check elementary service feature has also been successfully implemented as it checks through each service (A1…A5).
Changes to tasks/features	Throughout the week we found it difficult to allow the software to read the JSON input file. This made us temporarily turn to hard coding the input file into the software.
Current issues	-Upload and read input file: Our only issue this week was implementing a method to read JSON files. We found this particularly hard since the help from online websites were difficult to understand and implement.
Fixed issues	-After our meeting this week with Maria, our group found out that we were on the wrong track and misunderstood the instructions. Due to this we fell behind in our work and had to catch up. We fixed this issue by focusing our time and attention to the important tasks. -Haven't started creating our software when we were supposed to complete the features: Upload and read input file, check elementary service and generate L1. This issue was fixed since 3 out of the 4 group members focused on coding the software. -Haven't started documentation since we misunderstood Maria's instructions. One of our group members started working on documentation in order to catch up.
Current tasks (started working on)	-Upload and read input file -Find strongly connected service -Project charter -System design documentation
Future tasks	-Create software -Generate L1 -Generate L2 -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

Individual contributions (incl. timesheets):

Name	Contribution/Summary
Simon Gor	This week I focused on coding the software which was a challenge since I haven't coded in java for over a year. Once I refreshed my java knowledge, I turned my attention to implementing the 'upload and read input file' feature which became an issue as a lot of methods online were difficult to implement. In the end I passed this task to Yang and Zohoor and moved my focus onto check elementary service and strongly connected service.
Zohoor Yahya	This week Zohoor continued with developing and fixing the software. She spent the majority of her time getting the software to read the JSON file, in the end she hard coded the services, inputs and outputs as a temporary solution. Once this was done, Zohoor used my 'check elementary service' code and implemented it into the software. Even though many adjustments had to be made to the 'check elementary service code', Zohoor got it to work.
Yang Deng	In the early parts of the week, Yang spent his time learning more about JSON files and certain libraries that allow java to interact with JSON file format. He also attempted the 'upload and read input file' feature which he nearly got to work but had many issues with it. On the other hand Yang wrote the input file in JSON file format to use with our software.
Deok Kim	This week Deok attempted to understand the coding behind our software but struggled to make sense of it. In the end I spoke to him and asked him to start working on the project charter. He spent a significant amount of time creating the charter layout, writing the project purpose and project objectives.

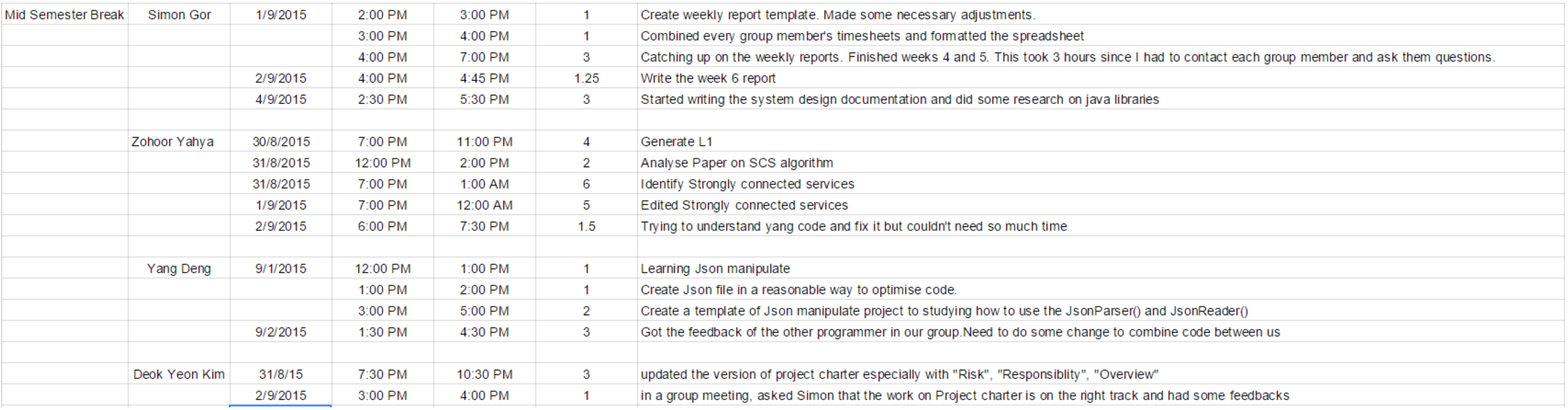


6	Simon Gor	24/8/15	11:00 AM	11:45 AM	0.75	Issue with eclipse software. Researched and fix the error that was appearing
			11:45 AM	1:00 PM	1.25	Refreshing java knowledge and researching on how to open files in java
			1:30 PM	3:00 PM	1.5	Implement JFileChooser to open a file. (Along the way we came across some errors)
			3:00 PM	4:30 PM	1.5	Research on how to read json files in Java
			5:00 PM	6:00 PM	1	Trying to understand how to code the "check elementary services" feature
		25/8/15	3:30 PM	5:00 PM	1.5	Learning the gson library through exmaples on the internet and attempted to make the program read .json files
			5:00 PM	5:45 PM	0.75	Updated Yang on the changes I've made to the program and wen't through some gson information since he isn't familiar with it
			5:45 PM	6:30 PM	0.75	Explained the "elementary service" logic to Kevin in order to give him some work
		26/8/15	11:30 AM	1:00 PM	1.5	Recieved updated work from Zohoor. Contained an error and attempted to fix it in order to run the code.
			2:30 PM	3:30 PM	1	Wrote check elementary service code
			3:30 PM	5:30 PM	2	Analysing strongly connected service and trying to workout how to write the code
		27/8/15	11:15 AM	12:00 PM	0.75	Group meeting with Maria and updated her with our progress
6	Zohoor	24/8/2015	9:00 PM	11:30 PM	2.5	analyse and code some changes to read and load file
		25/8/2015	7:00 PM	8:00 PM	1	Tried to fix error in file parsing, could not resolve it due to bad file format.
			8:00 PM	11:00 PM	3	Wrote code to generate sample data for the example given in research paper. This will facilitate parallel development as well as it can be used for verification.
			11:00 PM	1:00 AM	2	Wrote code for checking elementary service.
		26/8/2015	5:00 PM	7:00 PM	2	fixing bugs
		27/8/2015	11:00 AM	12:00 PM	1	
6	Yang Deng	8/24/2015	11:00 AM	12:00PM	1	Choose between Json or XML , then choose the appropriate Json library
		8/25/2015	3:00 PM	5:00 PM	2	Learning Gson library
			7:00 PM	10:00 PM	3	Based on the analsis of level0 and level1 ,Write the sample json file and refine it
		8/26/2015	5:00 PM	7:00 PM	2	Analyse the code of Zohoor
		8/27/2015	11:00 AM	12:00 PM	1	Group meeting with Maria and change our misunderstanding about the project
6	Deok Yeon Kim	25/8/15	3:30 PM	5:45 PM	2.25	Trying catching up with the programming work
		25/8/15	5:45 PM	6:30 PM	0.75	understanding the logic that I was unsure from the explanation from Simon
		26/8/15	11:30 AM	1:00 PM	1.5	Trying catching up with the programming work
			2:30 PM	3:30 PM	1	tried to work on elementary services coding
			6:30 PM	9:30 PM	3	finding information on 'Project charter' and writing

Week	**Mid Semester Break**
Completed tasks/features	-Generate L1: This week we have implemented the ‘Generate L1’ feature which utilizes the ‘check elementary service’ feature and generates the data from this. -Identify strongly connected services: Once L1 has been generated, we have implemented the ‘Identify strongly connected service’ algorithm to check the L1 data and recognize the strongly connected services.
Changes to tasks/features	There were no changes too tasks/features this week
Current issues	-Combining code from two members: We are currently having issues with combining Yang’s ‘Upload and read input file’ code with Zohoor’s code. This is difficult since the two styles of code are different from each other and they aren’t compatible with each other.
Fixed issues	-Upload and read input file: Our only issue this week was implementing a method to read JSON files. We found this particularly hard since the help from online websites were difficult to understand and implement. Yang has successfully found a way to read each line in a JSON file and pull that data into our java application.
Current tasks (started working on)	-Implementing ‘Upload and read input file’ code -Project charter -System design documents
Future tasks	-Create software -Generate L2 -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

Individual contributions (incl. timesheets):

Name	Contribution/Summary
Simon Gor	This week I focus on catching up on the weekly reports and documentation in general. I managed to finish weeks 4-6 reports along with combining our group's timesheets. On the other hand I started working on the software design documentation which explains our java code in detail.
Zohoor Yahya	This week Zohoor put in a lot of work as she finished 'Generate L1', and 'Identify strongly connected services'. In order for her to complete these features she did a lot of research and analysis of algorithm. Also Zohoor attempted to implement Yang's 'upload and read JSON input file' feature but had some issues with it.
Yang Deng	This week Yang worked hard on the 'Upload and read JSON input file' feature which had many issues in the previous weeks. After doing further research on the GSON library, he finally got it working.
Deok Kim	This week Deok continued with the documentations and added some more content into the project charter. He worked on the following headings: Risk, Responsibility and Overview.

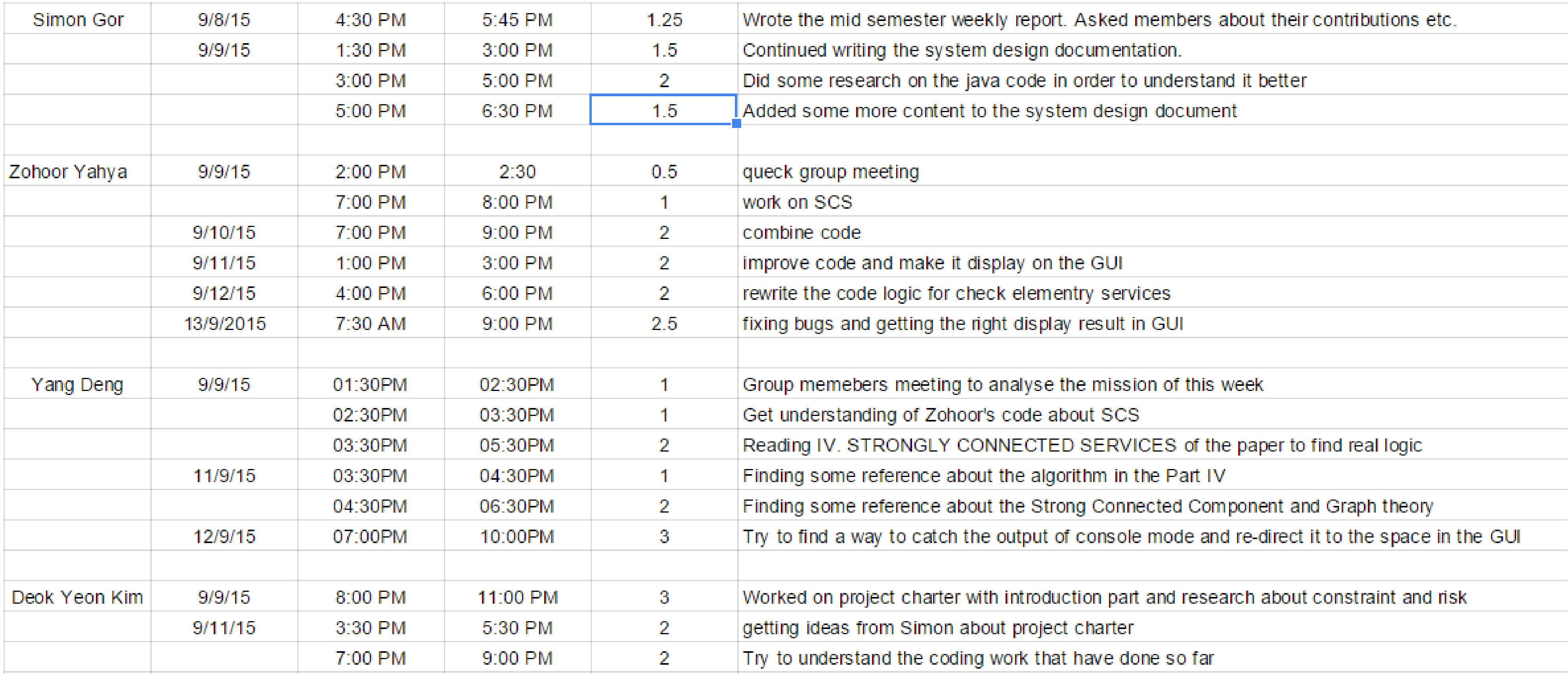


Mid Semester Break	Simon Gor	1/9/2015	2:00 PM	3:00 PM	1	Create weekly report template. Made some necessary adjustments.
			3:00 PM	4:00 PM	1	Combined every group member's timesheets and formatted the spreadsheet
			4:00 PM	7:00 PM	3	Catching up on the weekly reports. Finished weeks 4 and 5. This took 3 hours since I had to contact each group member and ask them questions.
		2/9/2015	4:00 PM	4:45 PM	1.25	Write the week 6 report
		4/9/2015	2:30 PM	5:30 PM	3	Started writing the system design documentation and did some research on java libraries
	Zohoor Yahya	30/8/2015	7:00 PM	11:00 PM	4	Generate L1
		31/8/2015	12:00 PM	2:00 PM	2	Analyse Paper on SCS algorithm
		31/8/2015	7:00 PM	1:00 AM	6	Identify Strongly connected services
		1/9/2015	7:00 PM	12:00 AM	5	Edited Strongly connected services
		2/9/2015	6:00 PM	7:30 PM	1.5	Trying to understand yang code and fix it but couldn't need so much time
	Yang Deng	9/1/2015	12:00 PM	1:00 PM	1	Learning Json manipulate
			1:00 PM	2:00 PM	1	Create Json file in a reasonable way to optimise code.
			3:00 PM	5:00 PM	2	Create a template of Json manipulate project to studying how to use the JsonParser() and JsonReader()
		9/2/2015	1:30 PM	4:30 PM	3	Got the feedback of the other programmer in our group.Need to do some change to combine code between us
	Deok Yeon Kim	31/8/15	7:30 PM	10:30 PM	3	updated the version of project charter especially with "Risk", "Responsiblity", "Overview"
		2/9/2015	3:00 PM	4:00 PM	1	in a group meeting, asked Simon that the work on Project charter is on the right track and had some feedbacks

Week	**7**
Completed tasks/features	-Display results in the GUI: This week we have implemented a method to display our results in a table format. This will make it easy to read and understand.
Changes to tasks/features	-Identify strongly connected service: Upon testing this feature, we found out that our code was incorrect as it was producing the wrong results. Ultimately our understanding of the algorithm was the reason why the results were wrong. This made us do further research and analysis in order to fix the issue.
Current issues	- Identify strongly connected service: As mentioned above, the algorithm we used and interpreted was incorrect since it produced results that didn't match the results represented in the paper. This is a major setback since originally we thought our first attempt was working.
Fixed issues	-Combining code from two members: We are currently having issues with combining Yang's 'Upload and read input file' code with Zohoor's code. This is difficult since the two styles of code are different from each other and they aren't compatible with each other. After many changes and bug fixes, we have successfully combined Yang's code with Zohoor.
Current tasks (started working on)	-Project charter -System design documents
Future tasks	-Create software -Generate L2 (now on hold) -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

Individual contributions (incl. timesheets):

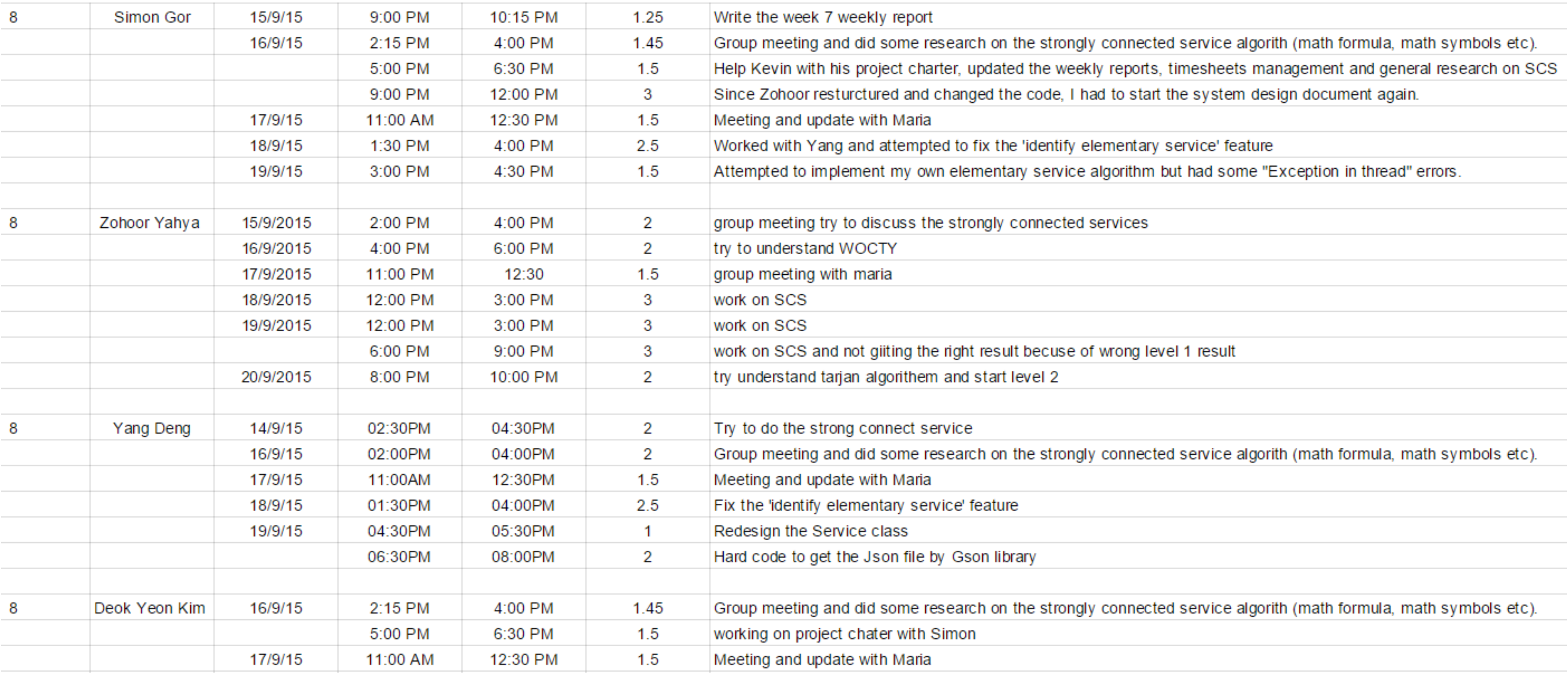
Name	Contribution/Summary
Simon Gor	This week I finished writing the mid semester weekly report and continued with the system design documentation. On the other hand I had to do some research and analysis on specific code that didn’t make sense to me.
Zohoor Yahya	This week Zohoor found out that the code for strongly connected service was incorrect since it produced the wrong results. She attempted to fix the issue but had many problems with it. Zohoor also improved the check elementary service code which had some bugs, successfully combined Yang’s code with her code and fixed the GUI so that the results could be displayed in a table format.
Yang Deng	After finding out that the strongly connected service code was incorrect, Yang did some further analysis on strongly connected service in order to help Zohoor. Yang also helped Zohoor combine their code and display the software results in a table format.
Deok Kim	This week Deok continued working on the project charter and research about the constraints and risks involved with this project. Since Deok has only been working on the documentation, he analyzed the current state of the code in order to see it’s progress.



Simon Gor	9/8/15	4:30 PM	5:45 PM	1.25	Wrote the mid semester weekly report. Asked members about their contributions etc.
	9/9/15	1:30 PM	3:00 PM	1.5	Continued writing the system design documentation.
		3:00 PM	5:00 PM	2	Did some research on the java code in order to understand it better
		5:00 PM	6:30 PM	1.5	Added some more content to the system design document
Zohoor Yahya	9/9/15	2:00 PM	2:30	0.5	queck group meeting
		7:00 PM	8:00 PM	1	work on SCS
	9/10/15	7:00 PM	9:00 PM	2	combine code
	9/11/15	1:00 PM	3:00 PM	2	improve code and make it display on the GUI
	9/12/15	4:00 PM	6:00 PM	2	rewrite the code logic for check elementry services
	13/9/2015	7:30 AM	9:00 PM	2.5	fixing bugs and getting the right display result in GUI
Yang Deng	9/9/15	01:30PM	02:30PM	1	Group memebers meeting to analyse the mission of this week
		02:30PM	03:30PM	1	Get understanding of Zohoor's code about SCS
		03:30PM	05:30PM	2	Reading IV. STRONGLY CONNECTED SERVICES of the paper to find real logic
	11/9/15	03:30PM	04:30PM	1	Finding some reference about the algorithm in the Part IV
		04:30PM	06:30PM	2	Finding some reference about the Strong Connected Component and Graph theory
	12/9/15	07:00PM	10:00PM	3	Try to find a way to catch the output of console mode and re-direct it to the space in the GUI
Deok Yeon Kim	9/9/15	8:00 PM	11:00 PM	3	Worked on project charter with introduction part and research about constraint and risk
	9/11/15	3:30 PM	5:30 PM	2	getting ideas from Simon about project charter
		7:00 PM	9:00 PM	2	Try to understand the coding work that have done so far

Week	**8**
Completed tasks/features	-No completed tasks this week: This week every team member worked hard to fix the SCS (strongly connected service) and check elementary service features. We put in many hours but still couldn't get these features to work.
Changes to tasks/features	-Check elementary service: After our meeting with Maria, the check elementary service code wasn't complete as the algorithm only worked in some scenarios. This resulted us in making some adjustments to the code in order to account for every scenario.
Current issues	-Identify strongly connected service: Unfortunately this feature is still an issue since it's hard to get a grasp of the OWCTY (one way catch them young) algorithm. Many hours have been put into this algorithm without any success. -Check elementary service: This issue just came up this week. Yang has been assigned this task and is working diligently to accommodate every scenario so that this algorithm can run as intended.
Fixed issues	-There were no fixed issues this week.
Current tasks (started working on)	-Project charter -System design documents
Future tasks	-Create software -Generate L2 (now on hold) -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

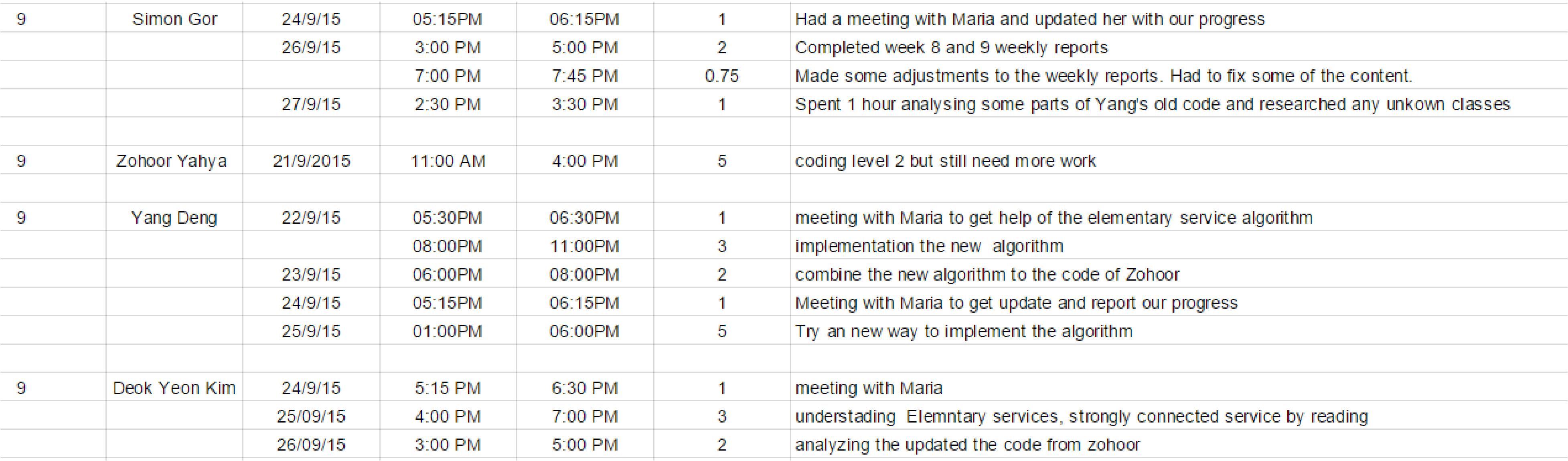
Name	Contribution/Summary
Simon Gor	This week I completed last weeks report (week 7) and put the majority of my time helping Yang and Kevin with their assigned tasks. In terms of my task with the system design documentation, I had to start fresh since Zohoor made major changes to her code structure which looked completely different to the original code I was working from.
Zohoor Yahya	This week Zohoor’s task was to work on the strongly connected service issue. She spent a lot of time trying to understand the algorithm but found the WOCTY algorithm too hard. In the end she opted to learn the Tarjan algorithm which will hopefully open the way for completing strongly connected services.
Yang Deng	Yang’s task for this week was to continue working on the ‘identify elementary service’ feature. Initially he tried to fix the strongly connected service feature but had no success. After working on ‘identify elementary service’ he found out that input JSON file didn’t contain the correct input data and therefore he had to recreate JSON file.
Deok Kim	This week Deok helped out a with the strongly connected service algorithm through doing some research. He also worked with me to fix some issues he had in his project charter.



8	Simon Gor	15/9/15	9:00 PM	10:15 PM	1.25	Write the week 7 weekly report
		16/9/15	2:15 PM	4:00 PM	1.45	Group meeting and did some research on the strongly connected service algorith (math formula, math symbols etc).
			5:00 PM	6:30 PM	1.5	Help Kevin with his project charter, updated the weekly reports, timesheets management and general research on SCS
			9:00 PM	12:00 PM	3	Since Zohoor resturctured and changed the code, I had to start the system design document again.
		17/9/15	11:00 AM	12:30 PM	1.5	Meeting and update with Maria
		18/9/15	1:30 PM	4:00 PM	2.5	Worked with Yang and attempted to fix the 'identify elementary service' feature
		19/9/15	3:00 PM	4:30 PM	1.5	Attempted to implement my own elementary service algorithm but had some "Exception in thread" errors.
8	Zohoor Yahya	15/9/2015	2:00 PM	4:00 PM	2	group meeting try to discuss the strongly connected services
		16/9/2015	4:00 PM	6:00 PM	2	try to understand WOCTY
		17/9/2015	11:00 PM	12:30	1.5	group meeting with maria
		18/9/2015	12:00 PM	3:00 PM	3	work on SCS
		19/9/2015	12:00 PM	3:00 PM	3	work on SCS
			6:00 PM	9:00 PM	3	work on SCS and not giiting the right result becuse of wrong level 1 result
		20/9/2015	8:00 PM	10:00 PM	2	try understand tarjan algorithem and start level 2
8	Yang Deng	14/9/15	02:30PM	04:30PM	2	Try to do the strong connect service
		16/9/15	02:00PM	04:00PM	2	Group meeting and did some research on the strongly connected service algorith (math formula, math symbols etc).
		17/9/15	11:00AM	12:30PM	1.5	Meeting and update with Maria
		18/9/15	01:30PM	04:00PM	2.5	Fix the 'identify elementary service' feature
		19/9/15	04:30PM	05:30PM	1	Redesign the Service class
			06:30PM	08:00PM	2	Hard code to get the Json file by Gson library
8	Deok Yeon Kim	16/9/15	2:15 PM	4:00 PM	1.45	Group meeting and did some research on the strongly connected service algorith (math formula, math symbols etc).
			5:00 PM	6:30 PM	1.5	working on project chater with Simon
		17/9/15	11:00 AM	12:30 PM	1.5	Meeting and update with Maria

Week	**9**
Completed tasks/features	-No completed tasks this week: This week was a difficult week since we found out that Zohoor was sick and had some health issues. Unfortunately she will be away for two weeks which will put the strongly connected service feature on a longer hold.
Changes to tasks/features	-There were no changes to tasks/features this week.
Current issues	-Identify strongly connected service: Same situation as last week since Zohoor got sick and this will be on hold until she gets better. -Check elementary service: Our current issue with check elementary is applying our knowledge of the feature to code. Yang who is predominantly working on this feature understands how the check elementary service feature is supposed to work but he finds it very difficult to translate his knowledge to code.
Fixed issues	-Check elementary service: Yang managed to fix bits and pieces of his algorithm but couldn't fully complete the algorithm.
Current tasks (started working on)	-Project charter -System design documents
Future tasks	-Create software -Identify strongly connected services (now on hold) -Generate L2 (on hold) -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

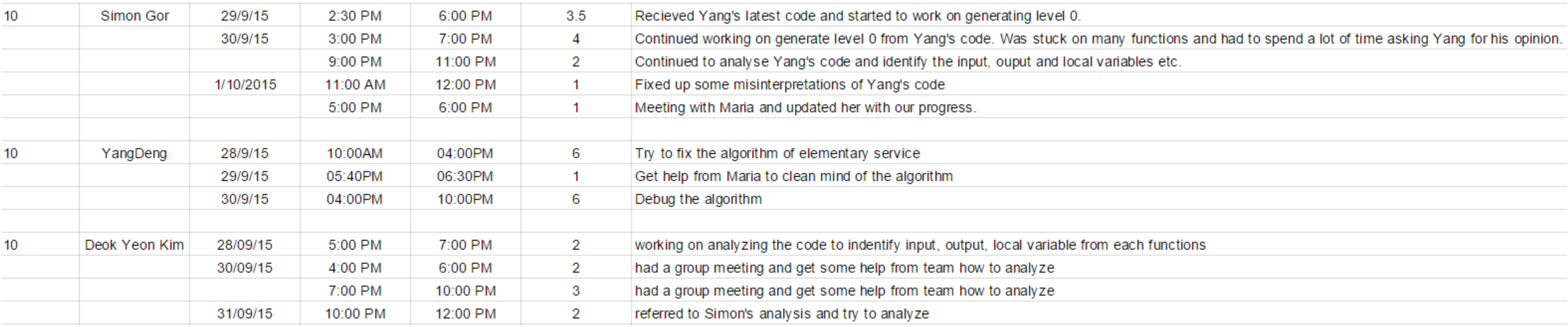
Name	Contribution/Summary
Simon Gor	This week I couldn’t spend as much time on the project due to other subject assignments. From what I could fit in, I managed to catch up on weeks 8 and 9 reports and spent an hour analysing Yang’s old code for the generate L0 from code feature.
Zohoor Yahya	This week Zohoor’s was sick and could only do some work on Monday. She spent a significant amount of time coding generate level 2 feature which is still in progress.
Yang Deng	Yang’s continued from last week and worked on the check elementary service feature. He managed to fix some parts of the algorithm and tried to implement those changes to the software. Throughout the week he was in contact with Maria for some help on particular issues with the algorithm.
Deok Kim	This week Deok had to catch up with his understanding of the algorithms in order to help out with the generate L0 from code. He also analysed Zohoor’s latest code in order to also help him understand the algorithm.



9	Simon Gor	24/9/15	05:15PM	06:15PM	1	Had a meeting with Maria and updated her with our progress
		26/9/15	3:00 PM	5:00 PM	2	Completed week 8 and 9 weekly reports
			7:00 PM	7:45 PM	0.75	Made some adjustments to the weekly reports. Had to fix some of the content.
		27/9/15	2:30 PM	3:30 PM	1	Spent 1 hour analysing some parts of Yang's old code and researched any unkown classes
9	Zohoor Yahya	21/9/2015	11:00 AM	4:00 PM	5	coding level 2 but still need more work
9	Yang Deng	22/9/15	05:30PM	06:30PM	1	meeting with Maria to get help of the elementary service algorithm
			08:00PM	11:00PM	3	implementation the new algorithm
		23/9/15	06:00PM	08:00PM	2	combine the new algorithm to the code of Zohoor
		24/9/15	05:15PM	06:15PM	1	Meeting with Maria to get update and report our progress
		25/9/15	01:00PM	06:00PM	5	Try an new way to implement the algorithm
9	Deok Yeon Kim	24/9/15	5:15 PM	6:30 PM	1	meeting with Maria
		25/09/15	4:00 PM	7:00 PM	3	understading Elemntary services, strongly connected service by reading
		26/09/15	3:00 PM	5:00 PM	2	analyzing the updated the code from zohoor

Week	**10**
Completed tasks/features	-Check elementary service: Yang has made some positive progress with the check elementary service feature. He has essentially completed the feature apart from a couple of bug which will be done within the next two days.
Changes to tasks/features	-There were no changes to tasks/features this week.
Current issues	-Identify strongly connected service: Same situation as last week since Zohoor got sick and this will be on hold until she gets better. -Check elementary service: There is currently a small issue with the algorithm when it checks each output data, sometimes it produces the incorrect results. Fortunately this is something small and can be fixed by the end of this week.
Fixed issues	-Check elementary service: The algorithm behind this feature is looking more promising. Apart from a few bugs, the main structure has been fixed and is 90% complete.
Current tasks (started working on)	-Project charter -System design documents
Future tasks	-Create software -Identify strongly connected service (on hold) -Generate L2 (on hold) -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Weekly reports -Issues and risks -Test plan -Peer review -Technical solution design documents -Learning outcomes

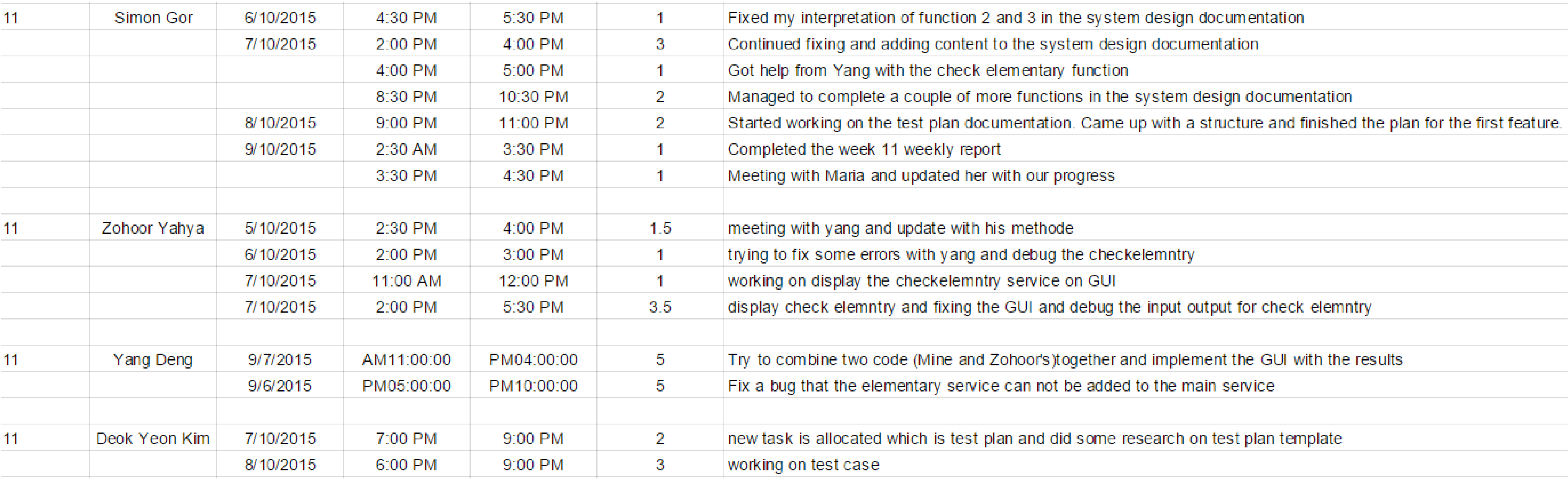
Name	Contribution/Summary
Simon Gor	This week we were informed by Maria that Kevin and I should start on the generate L0 from code. I spent the majority of this week analysing Yang's code and identifying the input, output, local variables and output dependencies. This took a very long time since it got confusing with particular functions. Most likely I've done the calculations wrong but gave it my best shot.
Zohoor Yahya	This week Zohoor's is still sick and couldn't do any work this week. She is planning to resume her work in week 11.
Yang Deng	Yang's determination and work ethic has helped him to essentially complete the check elementary service feature. He has finally completed the algorithm minus a few bugs but overall it's looking positive.
Deok Kim	This week Deok helped out with the analysis of Yang's code. Initially I helped him to understand how to find the input, output, local variables and output dependencies. Later he attempted to workout other functions and tried his best.



10	Simon Gor	29/9/15	2:30 PM	6:00 PM	3.5	Recieved Yang's latest code and started to work on generating level 0.
		30/9/15	3:00 PM	7:00 PM	4	Continued working on generate level 0 from Yang's code. Was stuck on many functions and had to spend a lot of time asking Yang for his opinion.
			9:00 PM	11:00 PM	2	Continued to analyse Yang's code and identify the input, ouput and local variables etc.
		1/10/2015	11:00 AM	12:00 PM	1	Fixed up some misinterpretations of Yang's code
			5:00 PM	6:00 PM	1	Meeting with Maria and updated her with our progress.
10	YangDeng	28/9/15	10:00AM	04:00PM	6	Try to fix the algorithm of elementary service
		29/9/15	05:40PM	06:30PM	1	Get help from Maria to clean mind of the algorithm
		30/9/15	04:00PM	10:00PM	6	Debug the algorithm
10	Deok Yeon Kim	28/09/15	5:00 PM	7:00 PM	2	working on analyzing the code to indentify input, output, local variable from each functions
		30/09/15	4:00 PM	6:00 PM	2	had a group meeting and get some help from team how to analyze
			7:00 PM	10:00 PM	3	had a group meeting and get some help from team how to analyze
		31/09/15	10:00 PM	12:00 PM	2	referred to Simon's analysis and try to analyze

Week	**11**
Completed tasks/features	-Check elementary service: Yang has managed to fix any existing issues with this feature.
Changes to tasks/features	-Check elementary service: Some small issues popped up (e.g. results were duplicated in the GUI) which made us modify and change some functions.
Current issues	-Identify strongly connected service: -Implementing check elementary service: Earlier this week we had an issue with implementing the check elementary service feature into Zohoor's code. This was particularly confusing since we had to create new functions/local variables in order to get both codes to run.
Fixed issues	-Identify strongly connected service: Same situation as last week since Zohoor got sick and this will be on hold until she gets better. Zohoor has rejoined our group after taking some time off due to her health issues. She started working on -Check elementary service: There is currently a small issue with the algorithm when it checks each output data, sometimes it produces the incorrect results. Fortunately this is something small and can be fixed by the end of this week. Yang has successfully eradicated all of the known small issues within this algorithm. Further testing this week will reveal if ALL issues have been fixed.
Current tasks (started working on)	-Project charter -System design documents -Test plan -Identify strongly connected service -Generate L2
Future tasks	-Create software -Annotate RCP (optional) -Generate L3 (optional) -Check property -Generate L0 from code -Write documentations -Issues and risks -Peer review -Technical solution design documents -Learning outcomes

Name	Contribution/Summary
Simon Gor	After our meeting in week 10, Maria advised us to start finishing our documentation. Hence this week I've been updating and completing more functions in the system design documentation. It has been taking this long since Zohoor and Yang are constantly updating their code. I've also managed to start the test plan and finish the first feature as well as complete week 11 weekly report.
Zohoor Yahya	This week Zohoor has rejoined our group after taking 2 weeks off from her studies due to health reasons. The early part of the week Zohoor got some help from Yang who updated her with our progress. One she got up to speed with our progress, Zohoor fixed the issue where the check elementary service couldn't display in the GUI.
Yang Deng	This week Yang worked on combining his code with Zohoor's. This took some time after many trial and errors but finally got it to work. He also fixed some other bugs including a bug that couldn't add the check elementary service function to the main service.
Deok Kim	This week Deok worked on a test case which will be used for testing the check elementary feature. Apart from the test case he did some research on the test plan templates which will be used for our documentation.



11	Simon Gor	6/10/2015	4:30 PM	5:30 PM	1	Fixed my interpretation of function 2 and 3 in the system design documentation
		7/10/2015	2:00 PM	4:00 PM	3	Continued fixing and adding content to the system design documentation
			4:00 PM	5:00 PM	1	Got help from Yang with the check elementary function
			8:30 PM	10:30 PM	2	Managed to complete a couple of more functions in the system design documentation
		8/10/2015	9:00 PM	11:00 PM	2	Started working on the test plan documentation. Came up with a structure and finished the plan for the first feature.
		9/10/2015	2:30 AM	3:30 PM	1	Completed the week 11 weekly report
			3:30 PM	4:30 PM	1	Meeting with Maria and updated her with our progress
11	Zohoor Yahya	5/10/2015	2:30 PM	4:00 PM	1.5	meeting with yang and update with his methode
		6/10/2015	2:00 PM	3:00 PM	1	trying to fix some errors with yang and debug the checkelemntry
		7/10/2015	11:00 AM	12:00 PM	1	working on display the checkelemntry service on GUI
		7/10/2015	2:00 PM	5:30 PM	3.5	display check elemntry and fixing the GUI and debug the input output for check elemntry
11	Yang Deng	9/7/2015	AM11:00:00	PM04:00:00	5	Try to combine two code (Mine and Zohoor's)together and implement the GUI with the results
		9/6/2015	PM05:00:00	PM10:00:00	5	Fix a bug that the elementary service can not be added to the main service
11	Deok Yeon Kim	7/10/2015	7:00 PM	9:00 PM	2	new task is allocated which is test plan and did some research on test plan template
		8/10/2015	6:00 PM	9:00 PM	3	working on test case

4 System Design Document

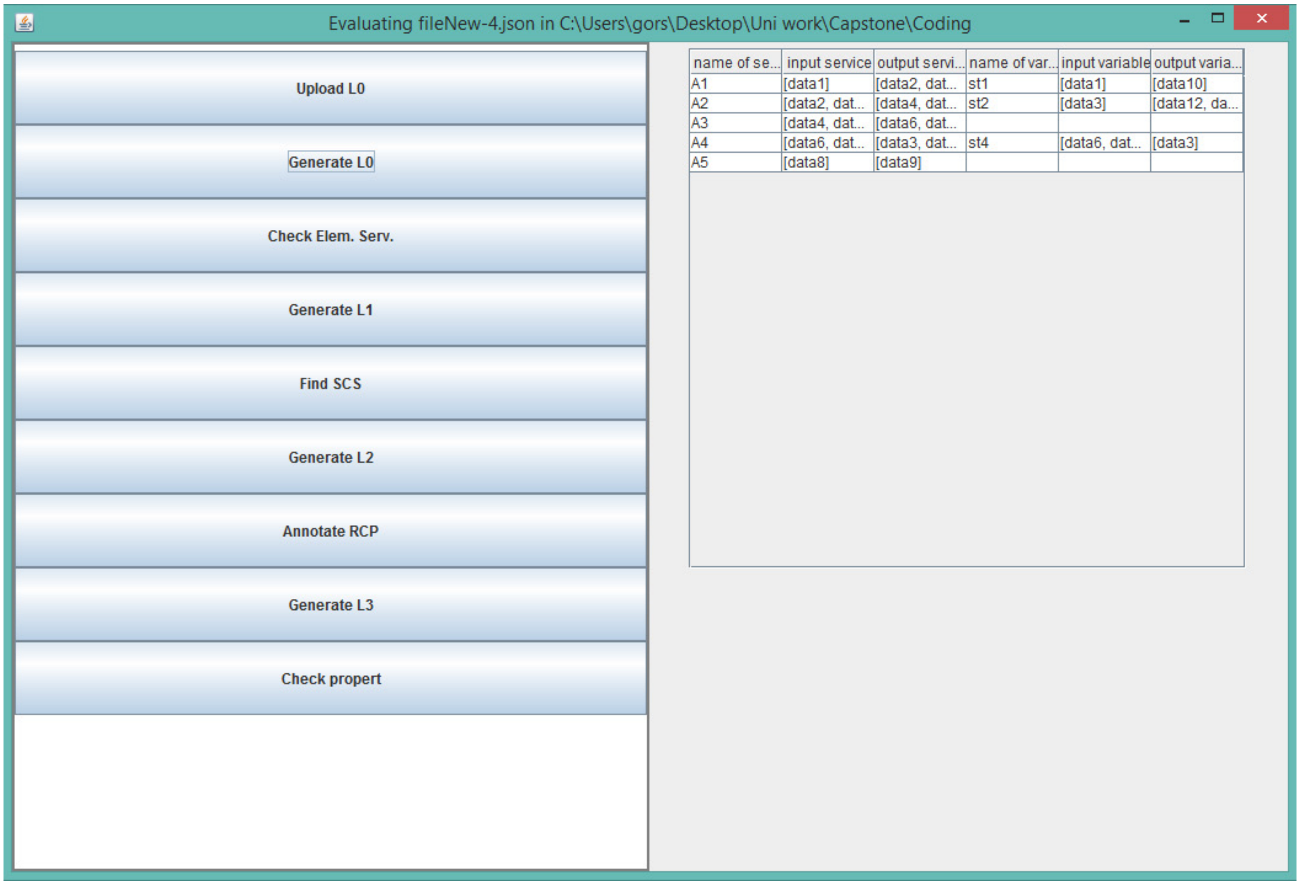
4.1 GUI (Graphical User Interface)

At the beginning of our project, our group made two decisions with the GUI:
1. Invest the majority of our time into the software features rather than the GUI
2. Use a clean interface that would display the software's results in a table
3. Locate the software buttons on the left hand side in a single column.

The first two decision were integral since this software was only intended for programmers, software testers etc which is why we didn't invest heavily in creating a beautiful/intuitive GUI. Also displaying the results in a table will help our intended audience to find and identify the results they are looking for. Meanwhile since this software only contains eight features, we allocated one button for each feature and didn't choose to place them in a row since rows are more for drop down menus that categories each feature.

The first design of this software contained buttons that were unnecessarily large and it affected the way the results were displayed. The columns in our table were too small and only displaying a small section of our results. The user would have to manually adjust the columns so that they can see the full results.

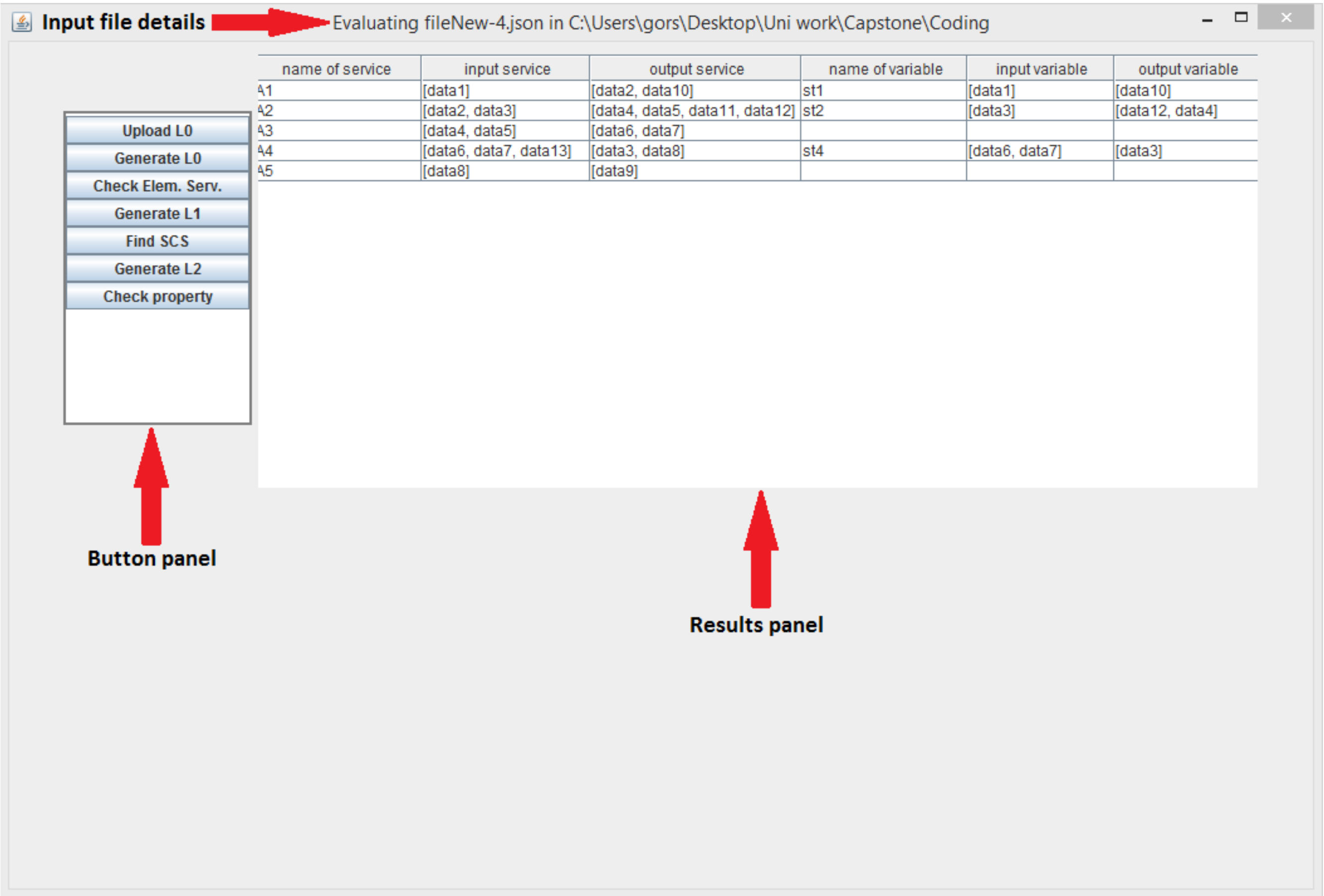
This first version of our GUI can be seen from below:



name of se...	input service	output servi...	name of var...	input variable	output varia...
A1	[data1]	[data2, dat...	st1	[data1]	[data10]
A2	[data2, dat...	[data4, dat...	st2	[data3]	[data12, da...
A3	[data4, dat...	[data6, dat...			
A4	[data6, dat...	[data3, dat...	st4	[data6, dat...	[data3]
A5	[data8]	[data9]			

After getting some advice we made adjustments to our buttons in order to allow breathing space for the results to display. But the drawback to this was our buttons were small and a little bit harder to click.

The second version of our GUI can be seen from below:

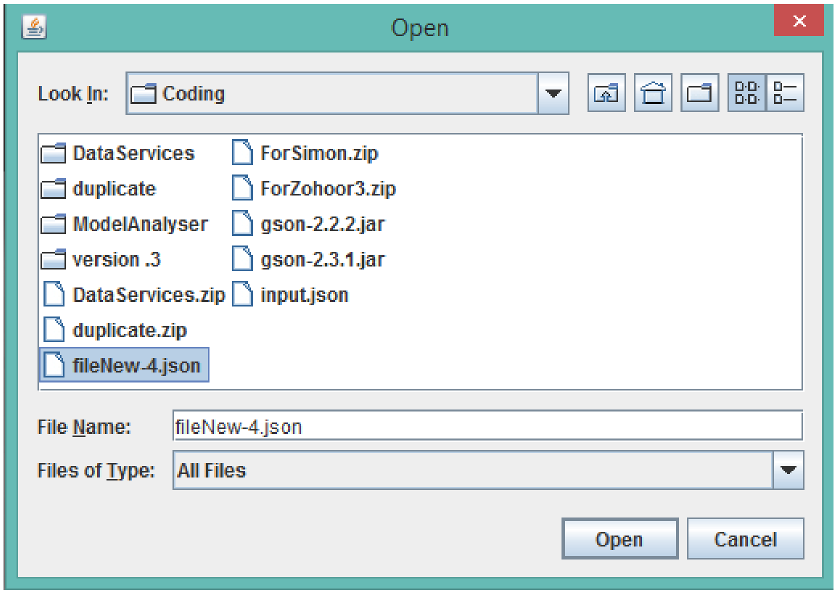


Evaluating fileNew-4.json in C:\Users\gors\Desktop\Uni work\Capstone\Coding

name of service	input service	output service	name of variable	input variable	output variable
A1	[data1]	[data2, data10]	st1	[data1]	[data10]
A2	[data2, data3]	[data4, data5, data11, data12]	st2	[data3]	[data12, data4]
A3	[data4, data5]	[data6, data7]			
A4	[data6, data7, data13]	[data3, data8]	st4	[data6, data7]	[data3]
A5	[data8]	[data9]			

4.2 Upload L0

The upload L0 feature is very simple since it utilizes a couple of functions such as JFileChooser, BufferReader and FileReader to perform the task of uploading the input file and reading the contents of the file. We've also incorporated the Google JSON library called Gson to also help us read the input JSON file through functions such as fromJson etc. The result from this function looks like this:

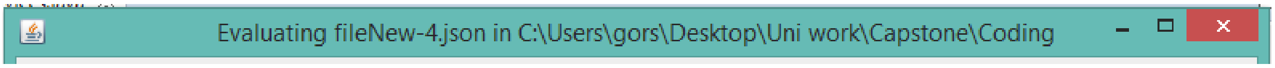


Look In: Coding

File Name: fileNew-4.json

Files of Type: All Files

In terms of the visual aspect, Upload L0 displays a simple line in the title bar displaying which file is being evaluated and from which location [Evaluating <file name> in <location>]. We chose to keep this separate from the results area since it was only specifically designed to display tables of data (our results).

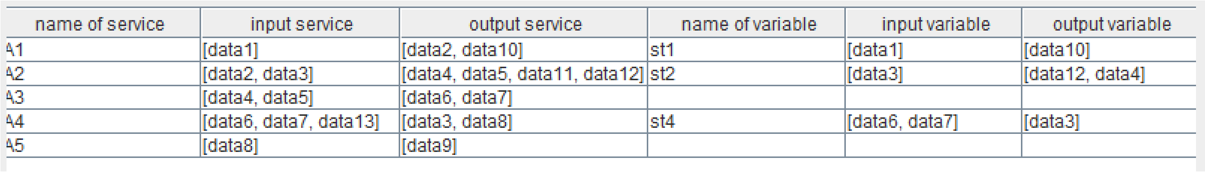


Evaluating fileNew-4.json in C:\Users\gors\Desktop\Uni work\Capstone\Coding

4.3 DISPLAY L0

The display L0 feature is linked with the Upload L0 feature by displaying the input file in the results panel as a table. We've made the decision to keep these two features separate since at times users will switch between features for example switching between Check elementary service and Display L0 in order to see the input and output of each subservice. Ultimately this will stop the file browser prompt from displaying each time (which can be quite annoying).

This feature is designed based off a single function which utilizes the Upload L0 data and runs it through a single nested for loop. This for loop uses getters to collect the data and displays them in the through specifying the location (row index) of each data.



name of service	input service	output service	name of variable	input variable	output variable
A1	[data1]	[data2, data10]	st1	[data1]	[data10]
A2	[data2, data3]	[data4, data5, data11, data12]	st2	[data3]	[data12, data4]
A3	[data4, data5]	[data6, data7]			
A4	[data6, data7, data13]	[data3, data8]	st4	[data6, data7]	[data3]
A5	[data8]	[data9]			

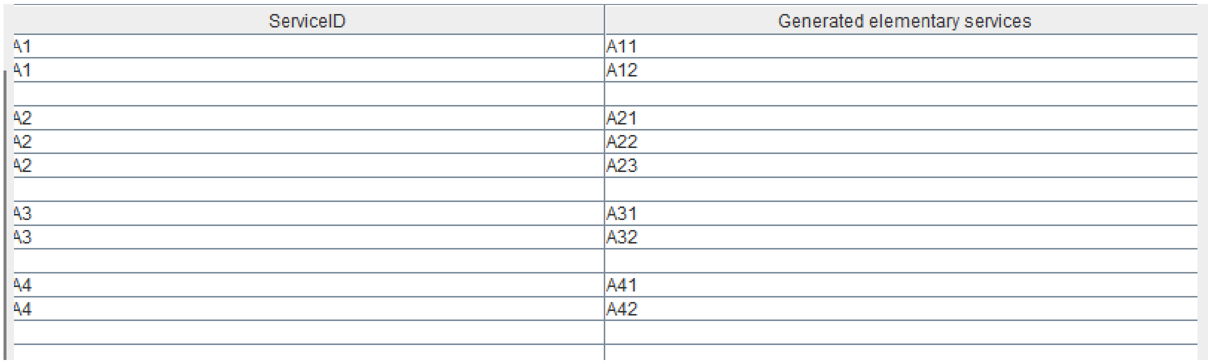
4.4 CHECK ELEMENTARY SERVICE

The check elementary service feature is intended to check the results from Display L0 feature and analyse the data based on two dot points from the paper "Towards Logical Architecture and Formal Analysis of Dependencies Between Services". These points state that:
1. An elementary service should have a single output channel.
2. All its output channels are correlated.
The purpose of the two dot points is to identify which services are elementary in order to create them into sub services.

Based from the two dot points above, we've designed check elementary function to check three different cases in order to meet every possible scenario.

- The first and simplest case checks if the service contains **only** one output. This case is created by a single IF loop that identifies the output variable size is equal (==) to one. Lastly if this case is true then the first dot point (an elementary service should have a single output channel) is satisfied and therefore is an elementary service.
- The second case initially looks at if service contains no local variable. The reason why we look at the local variable is because it's an important variable in the decomposition/separation process when creating the elementary service. Lastly a while loop checks if there is at least one output in order to identify if it's an elementary service and run's the decomposition algorithm to create the subservices.
- The third and final case also initially checks the local variable but this time if the local variable is greater than 0. When the local variable is greater than 0, the algorithm will firstly identify the output data from the services (must contain at least one output) and then creates subservices by pairing all the outputs who use the same local variable into a single subservice.

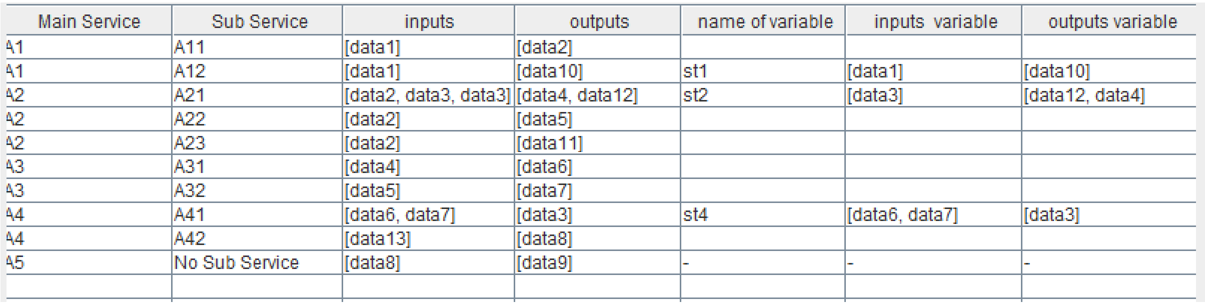
Once all of these calculations are done, the results are then display in a single two column table which shows the generated subservices created from their original service (as seen below).



ServiceID	Generated elementary services
A1	A11
A1	A12
A2	A21
A2	A22
A2	A23
A3	A31
A3	A32
A4	A41
A4	A42

4.5 GENERATE L1 (NEEDS FIXING)

In order to generate our findings from check elementary service, we've created a detailed and structured table that displays the main service, sub service, inputs, ouputs, name of variable, inputs variable and outputs variable. The code behind this table isn't a complex algorithm but rather simply getting stored values and displaying them in a table column.



Main Service	Sub Service	inputs	outputs	name of variable	inputs variable	outputs variable
A1	A11	[data1]	[data2]			
A1	A12	[data1]	[data10]	st1	[data1]	[data10]
A2	A21	[data2, data3, data3]	[data4, data12]	st2	[data3]	[data12, data4]
A2	A22	[data2]	[data5]			
A2	A23	[data2]	[data11]			
A3	A31	[data4]	[data6]			
A3	A32	[data5]	[data7]			
A4	A41	[data6, data7]	[data3]	st4	[data6, data7]	[data3]
A4	A42	[data13]	[data8]			
A5	No Sub Service	[data8]	[data9]	-	-	-

4.6 FIND STRONGLY CONNECTED SERVICE

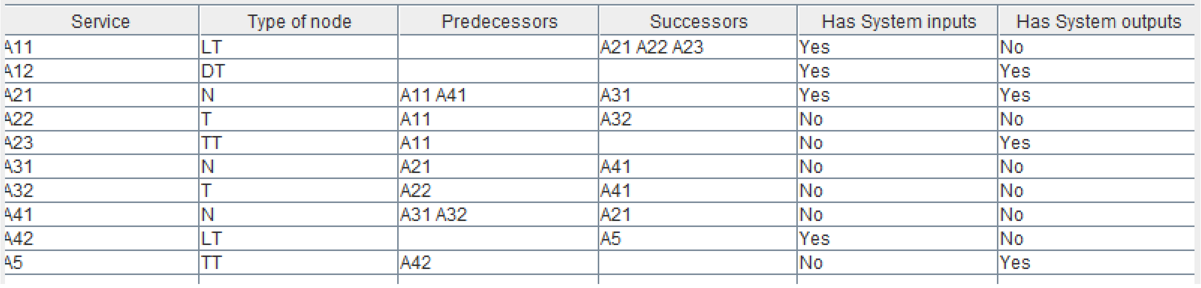
The Find strongly connected service feature is designed to use the data produced from Generate L1 and follow the three procedures represented in Maria's paper. The three summarised procedures are:

1. Identify DT services and delete these vertices.
2. Apply the recursive elimination technique, in this case we used Tarjan's algorithm to remove LT and TT services. [Note: refer to 8.2 for explanation as to why we used Tarjan algorithm and how it works]
3. Turn the removed services (DT, LT and TT) into a single service on L2. Then apply the existing algorithms. In this case we applied the Tarjan algorithm.

In terms of the code structure, we've designed the Find Strongly Connected Service into one function that utilises two separate classes in order to perform and complete step 1 and parts of step 2-3 (the remaining completion of steps 2-3 will be done in the next feature "Generate L2"). The reason why we used one function and two classes was based on the large number of processes it took to identify the nodes, successors, predecessors etc. It also helps to keep track of everything by splitting the functionality into three separate classes rather than one big class.

In regards to the function and a small overview of how we designed it, it is called `identifyStronglyConnectedServices()` which utilizes the two classes called `Service()` and `ServiceNode()`. In total `identifyStronglyConnectedServices()` contains three main processes that calculates the node type and whether the system has input or output. The main use for `Service()` class is to extract all of the Level L1 data such as output, input, service name which are stored in lists (arrayList, hashMap etc). Meanwhile the `ServiceNode()` calculates the predecessors and successors which is passed to the main class and use in its calculations.

Once all of the calculations are completed, the results are displayed in a 6 column table that displays the service name, type of node, predecessors, successors, if the system has inputs and if the system has outputs. An example of this table can be seen from below:



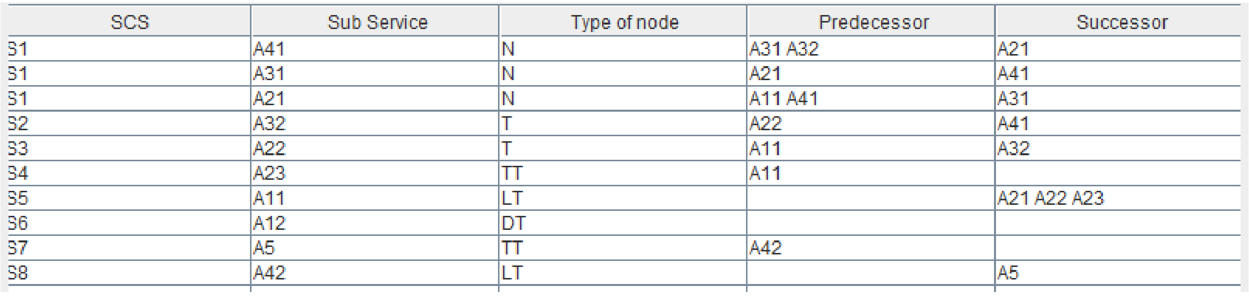
Service	Type of node	Predecessors	Successors	Has System inputs	Has System outputs
A11	LT		A21 A22 A23	Yes	No
A12	DT			Yes	Yes
A21	N	A11 A41	A31	Yes	Yes
A22	T	A11	A32	No	No
A23	TT	A11		No	Yes
A31	N	A21	A41	No	No
A32	T	A22	A41	No	No
A41	N	A31 A32	A21	No	No
A42	LT		A5	Yes	No
A5	TT	A42		No	Yes

4.7 GENERATE L2

The Generate L2 feature is an extension from the Find Strongly Connected Service feature. Since the Find Strongly Connected Service feature only satisfies steps 1 and partly steps 2-3, Generate L2 completes the rest of the calculations by implementing the elimination technique and service removal through the use of Tarjan's algorithm.

The code design of Generate L2 contains a total of two functions called `generateLevel2()` and `strongConnect()`. Both of these functions use segments of Tarjan's algorithm to detect whether the service is Strongly Connected and groups them according to the algorithm. Lastly the one design change in Generate L2 is the use of N. This string was selected by Zohoor to identify which nodes are neither T, TT, LT nor DT.

Once Tarjan's algorithm has been implemented, the results are displayed in a 5 column table with the headings SCS (strongly connected service), Sub Service, Types of node, predecessor and successor.



SCS	Sub Service	Type of node	Predecessor	Successor
S1	A41	N	A31 A32	A21
S1	A31	N	A21	A41
S1	A21	N	A11 A41	A31
S2	A32	T	A22	A41
S3	A22	T	A11	A32
S4	A23	TT	A11	
S5	A11	LT		A21 A22 A23
S6	A12	DT		
S7	A5	TT	A42	
S8	A42	LT		A5

4.8 REMOTE COMPUTATION

The Remote Computation feature is designed to build a system architecture that is optimised for remote computation. To achieve this optimisation a dynamic checkbox alongside a series of process are implemented into the calculations. Here are the list of processes we've designed represented in paper:

1. Display a dynamic checkbox prompting the user to select which data are High Load.
2. The selected data is checked with the Level L2 system and determine which input and output service are High Load. E.g. data5 is the input for S2 and the output for S3. Please refer to Fig 4.11.4.
3. Once the services are determined as High Load, they are combined and stored as a new group. E.g. S2 and S3 from data 5 are combined [S2, S3] and is given a new group name -> [S2, S3] = RS2
4. The remaining services that aren't High Load are then also combined into a group just like step 3.
5. A new dynamic checkbox is prompting the user to select which sub services are High Performance.
6. The High Performance components are then matched with their corresponding High Load in step 3. This will eliminate any High Load services that don't correspond to any of the selected High Performance sub service.
7. The remaining High Performance services are grouped and displayed as the final result. This is the final optimised architecture (known as Level L3).

In terms of the code structure and it's design, Remote Computation is created through one main class called `RemoteComputation()` which utilizes the class called `StronglyConnectedService()`. The reason why it utilizes this class is because all the Level L2 data is stored here and since Remote Computation only uses data from Level L2, this was the best way to pass relevant data into the main class. Once the data is passed back to `RemoteComputation()`, this class will then run the algorithm created by Yang and complete the 7 processes as mention above.

4.9 CHECK PROPERTY

The check property feature is designed to check the relations between input and output data from a Level L2 system. This is important since it checks if there are missing or too many data which aren't in relation to each other. Meanwhile the check property also checks if a service is in relation to the input and output data. In total there are a total of 5 processes that take place in order to check property.

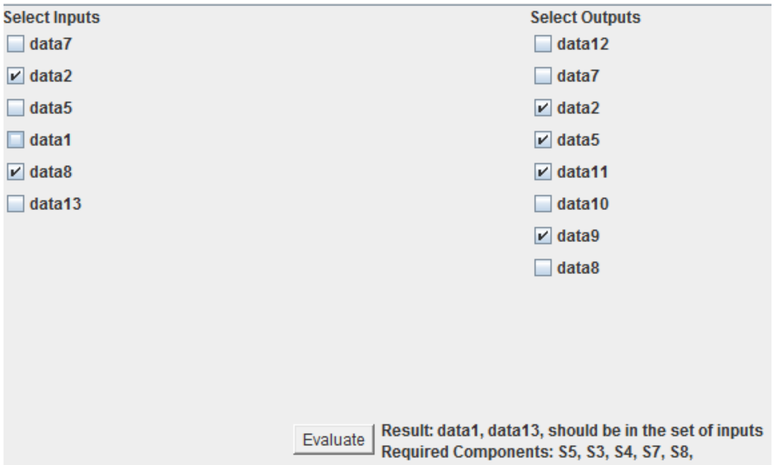
1. The user selects one or more input and output data from the dynamic checkboxes.
2. The selected output from step 1 is then checked to see which services in the Level L2 system are in relation. E.g. If data9 is selected then S8 and S7 are related based on Level L2 system [Fig 4.11.4].
3. Use the services found in step 2 and identify their input data. E.g. S8 has data13 as its input and S7 has data8 as its input.
4. Compare the input data selected from step 1 and the input data from step 3.
5. Display the final results which shows the input data that needs to be included/excluded and the required services.

When it came to the code design, the Check Property code is very long since it needs to check a lot of information in order to produce the desired results. In total the Check Property contains 9 functions with the majority of the functions checking and validating the Level L2 system.
A design summary of each function:

1. `checkProperty()` – This function is the longest of the 9. The first purpose of this function is to collect data of each strongly connected service and elementary service input and ouput values. This utilizes the `ServiceNode()` and `StronglyConnectedService()` classes in order to achieve this. The second and last purpose of this function is to display the GUI which includes the dynamic checkboxes and the final results.
2. `findRequiredInputs()` – This function uses the strongly connected information collected from the `checkProperty()` function and identifies whether the input is not duplicated.
3. `findSources()` – This function checks whether the output is a source also takes into consideration of duplicates.
4. `canReachOutput()` – This function checks if the service has an output that matches the final destination output.
5. `canReachInput()` – This function validates whether the dependency is correct and what we are looking for.
6. `dependencySearch()` – This function finds all dependencies in Level L2 and stores the values in a ArrayList.
7. `dependentSearch()` – Much like `dependencySearch()`, this function finds all of the dependent in Strongly connected service and stores the data in an ArrayList.

8. `getImmediateDependencies()` – This function uses the immediate dependencies and checks whether the source has an output that matches one of the targeted inputs.
9. `getImmediateDependents()` – This function uses the immediate dependents and checks whether the source has an input that matches one of the targeted outputs.

Once all of the functions and classes work cohesively, we've designed the results to display below the dynamic checkbox which can be seen in the example below.



Select Inputs

☐ data7
☑ data2
☐ data5
☐ data1
☑ data8
☐ data13

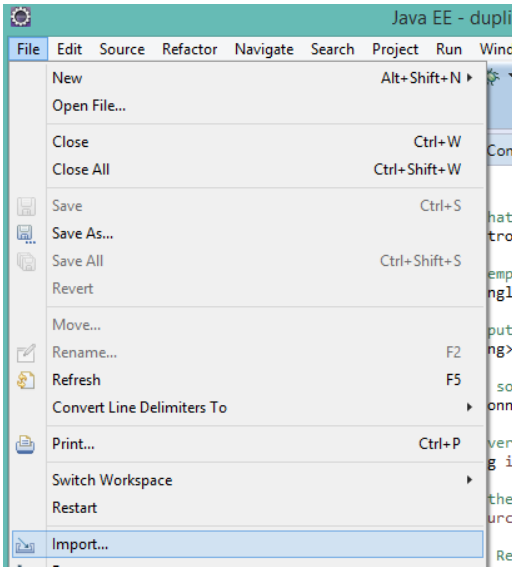
Select Outputs

☐ data12
☐ data7
☑ data2
☑ data5
☑ data11
☐ data10
☑ data9
☐ data8

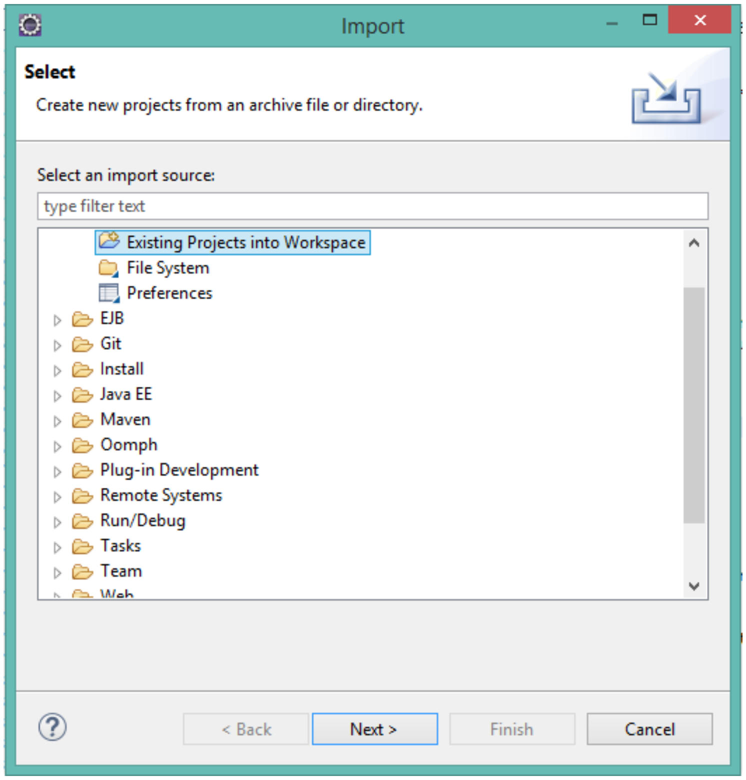
Result: data1, data13, should be in the set of inputs
Required Components: S5, S3, S4, S7, S8,

4.10 INSTRUCTIONS ON HOW TO USE ECLIPSE

Step 1: Click on File and select Import.



Step 2: A new prompt will appear called “Import”. In this window double click on “Existing projects in workspace”.

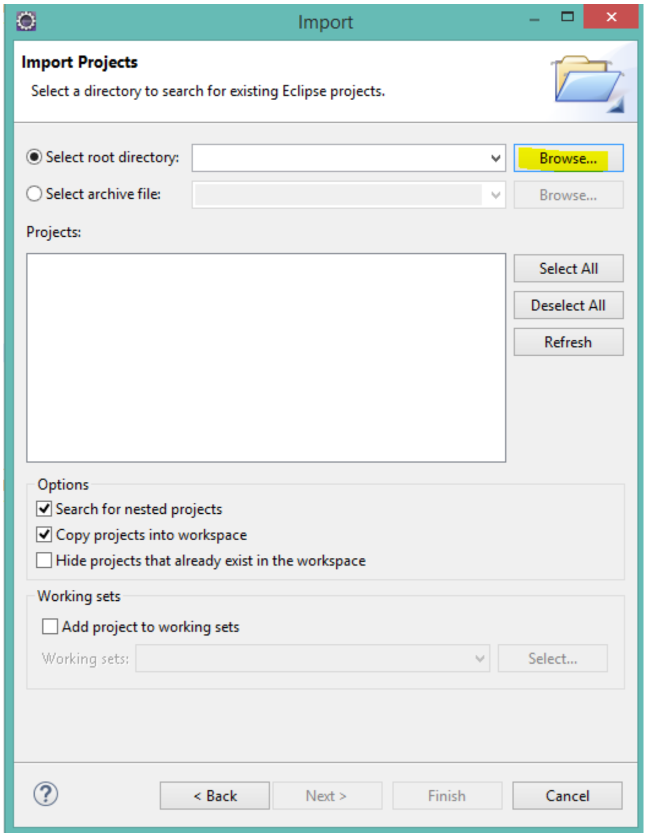

Import

Select

Create new projects from an archive file or directory.

Select an import source:

Step 3: Make sure “select root directory” is checked and click on browse.



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

Options

☑ Search for nested projects

☑ Copy projects into workspace

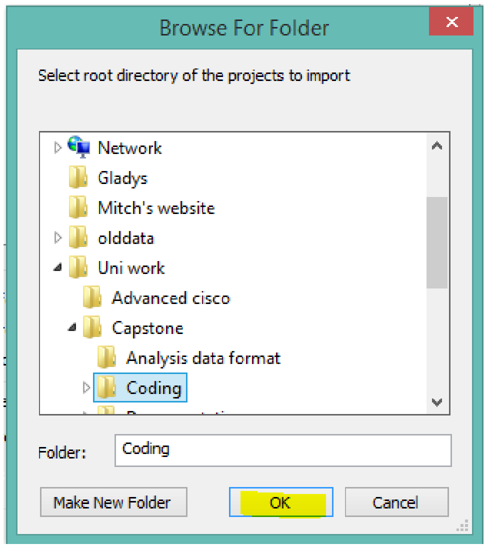
☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

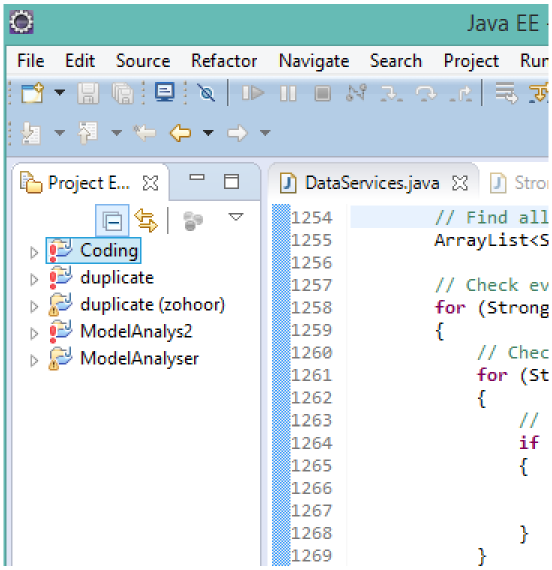
Step 4: Browse and find the location of the project folder and then click OK.



Select root directory of the projects to import

Folder: Coding

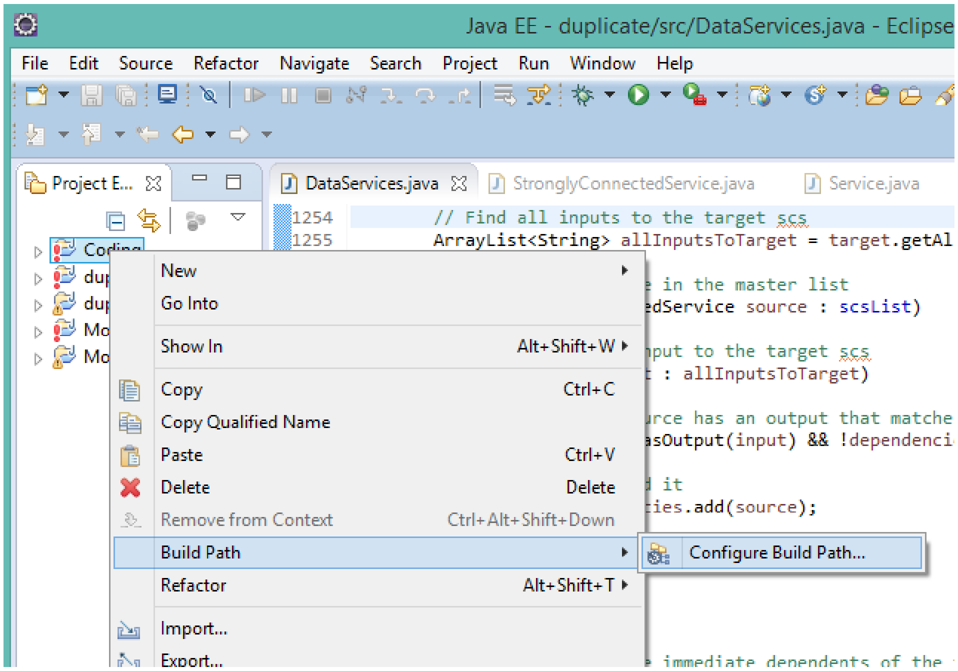
Step 5: In Eclipse you will see that the folder has been successfully imported in the left panel.



```
        // Find all
        ArrayList<S

        // Check ev
        for (Strong
        {
            // Chec
            for (St
            {
                //
                if
                {


                }
            }
```

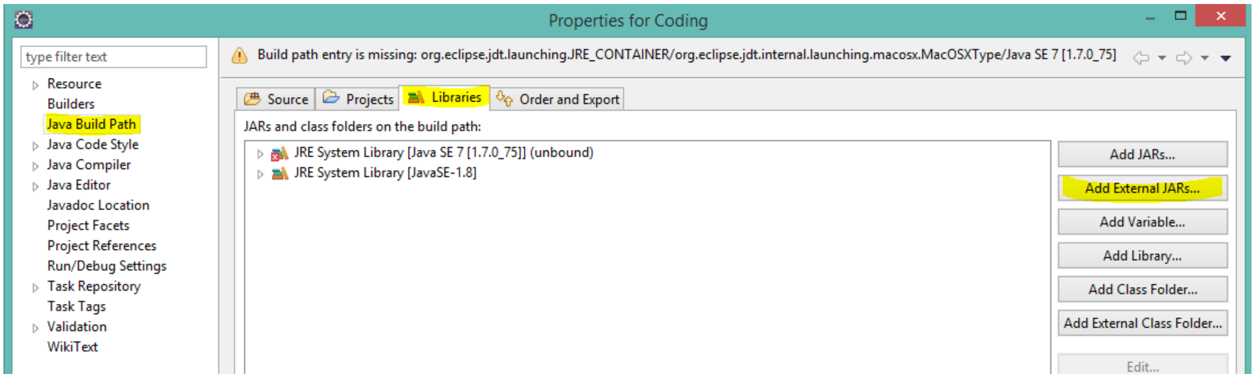
Step 6: Download gson-2.3.1.jar from http://mvnrepository.com/artifact/com.google.code.gson/gson/2.3.1 . This is the gson library which we will be using.

Step 6: Right click on the project folder and select “Build Path” then “Configure Build Path…”



```
        // Find all inputs to the target scs
        ArrayList<String> allInputsToTarget = target.getAl

e in the master list
edService source : scsList)

nput to the target scs
t : allInputsToTarget)

urce has an output that matche
asOutput(input) && !dependenci

d it
ies.add(source);


e immediate dependents of the
```

Step 7: A new window called "Properties for Coding" will display which will go directly to "Java Build Path" -> "Libraries". You will need to click on "Add External JARs…"

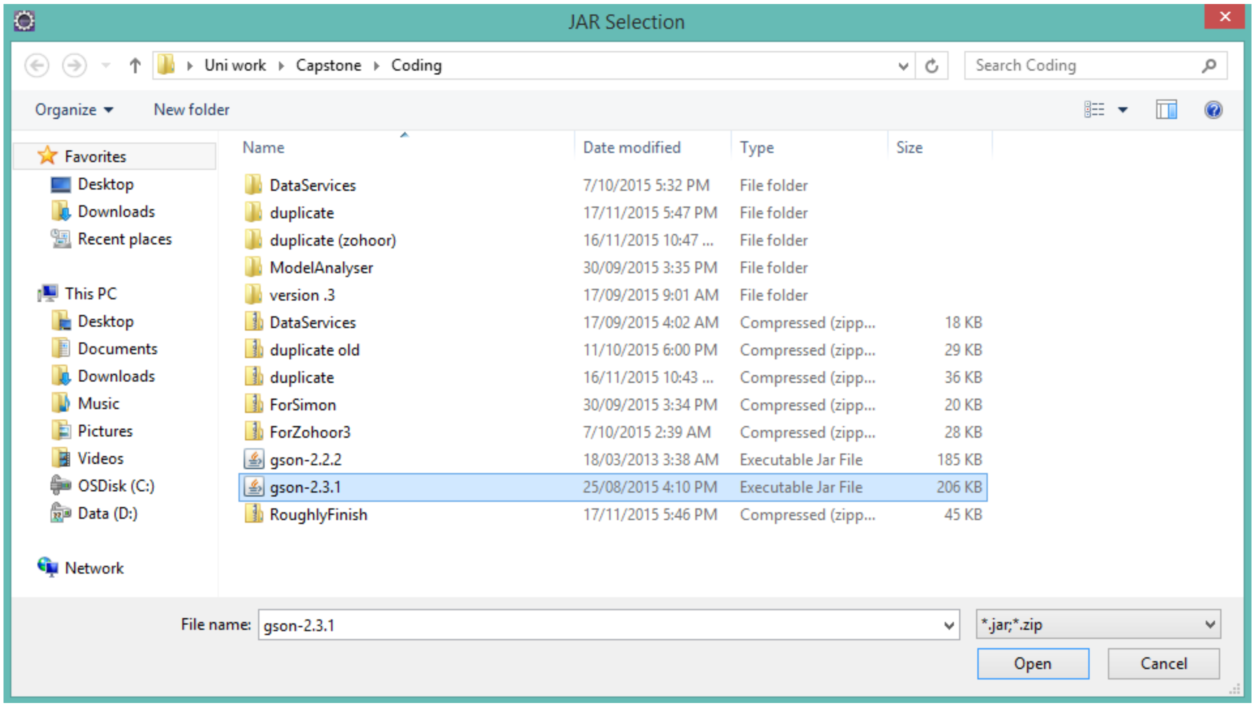


Build path entry is missing: org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.launching.macosx.MacOSXType/Java SE 7 [1.7.0_75]

JARs and class folders on the build path:

- JRE System Library [Java SE 7 [1.7.0_75]] (unbound)
- JRE System Library [JavaSE-1.8]

Step 8: Browse your folders and find the download gson-2.3.1.jar file. Once you have found it, select it and then click on Open.

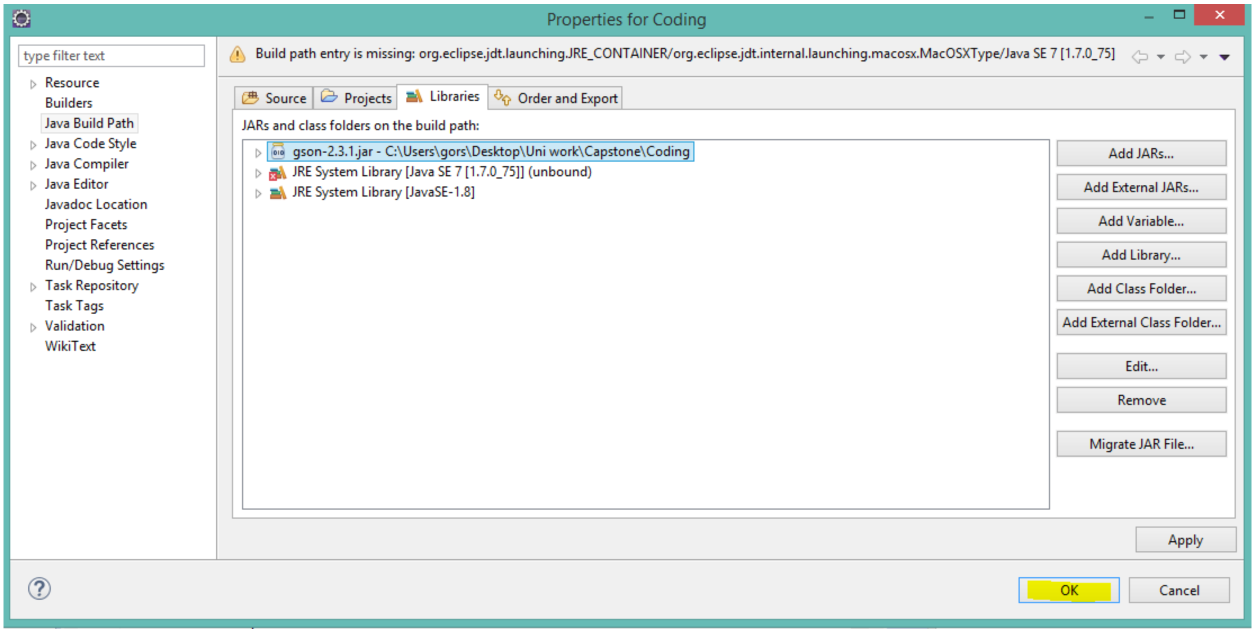


Name	Date modified	Type	Size
DataServices	7/10/2015 5:32 PM	File folder	
duplicate	17/11/2015 5:47 PM	File folder	
duplicate (zohoor)	16/11/2015 10:47 ...	File folder	
ModelAnalyser	30/09/2015 3:35 PM	File folder	
version .3	17/09/2015 9:01 AM	File folder	
DataServices	17/09/2015 4:02 AM	Compressed (zipp...	18 KB
duplicate old	11/10/2015 6:00 PM	Compressed (zipp...	29 KB
duplicate	16/11/2015 10:43 ...	Compressed (zipp...	36 KB
ForSimon	30/09/2015 3:34 PM	Compressed (zipp...	20 KB
ForZohoor3	7/10/2015 2:39 AM	Compressed (zipp...	28 KB
gson-2.2.2	18/03/2013 3:38 AM	Executable Jar File	185 KB
gson-2.3.1	25/08/2015 4:10 PM	Executable Jar File	206 KB
RoughlyFinish	17/11/2015 5:46 PM	Compressed (zipp...	45 KB

File name: gson-2.3.1

.jar;.zip

Step 9: This will bring you back to the “Properties for Coding” window. Now you simply click on OK.

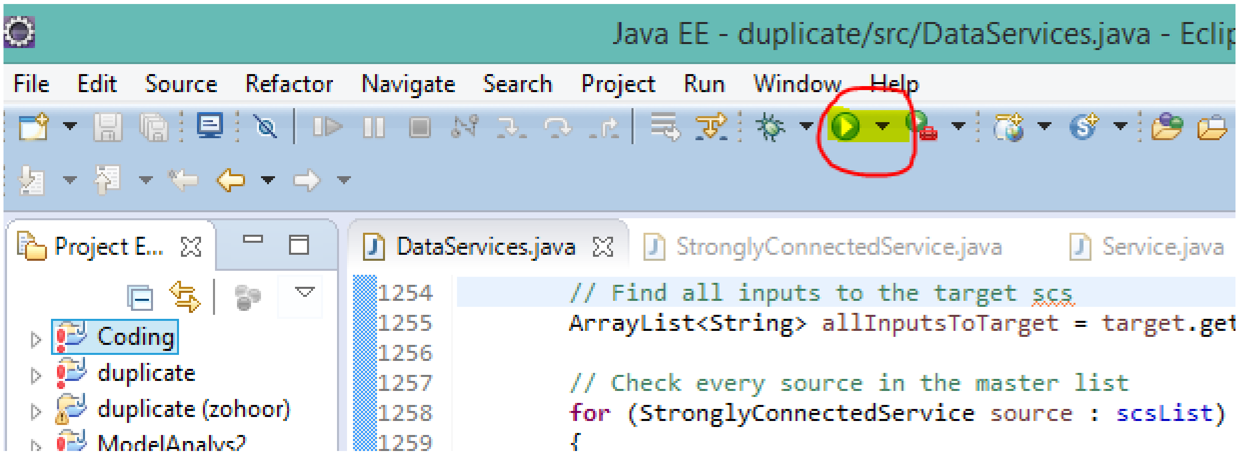


Build path entry is missing: org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.launching.macosx.MacOSXType/Java SE 7 [1.7.0_75]

JARs and class folders on the build path:

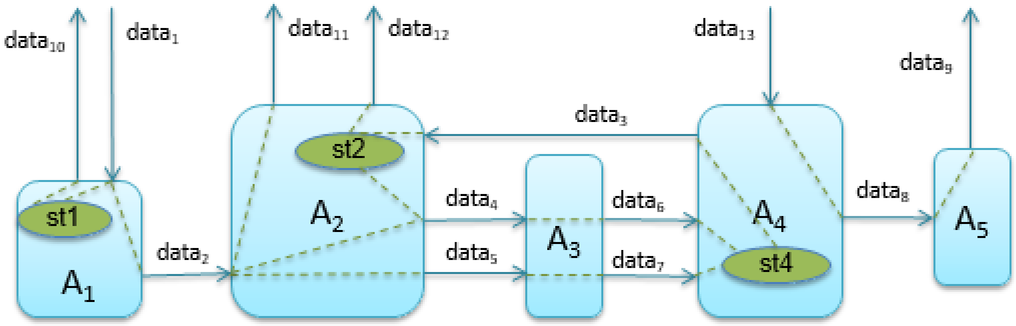
- gson-2.3.1.jar - C:\Users\gors\Desktop\Uni work\Capstone\Coding
- JRE System Library [Java SE 7 [1.7.0_75]] (unbound)
- JRE System Library [JavaSE-1.8]

Step 10: You have successfully configured the project and able to run code by pressing on the green arrow.

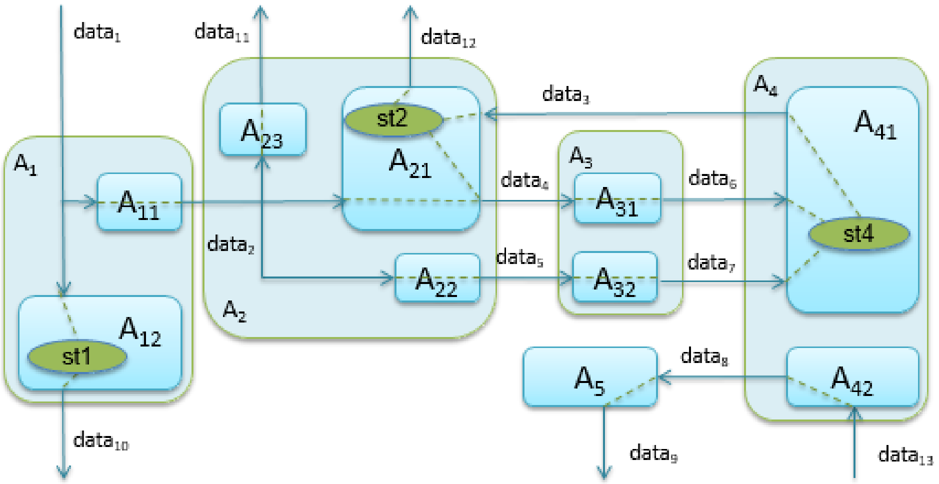


```
        // Find all inputs to the target scs
        ArrayList<String> allInputsToTarget = target.get

        // Check every source in the master list
        for (StronglyConnectedService source : scsList)
        {
```

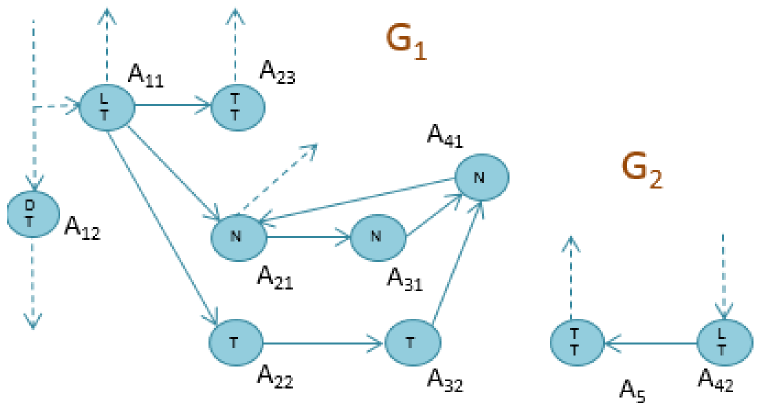
4.11 DIAGRAM OF SYSTEM WITH NEW NAMES



4.11.1 Level L0 system

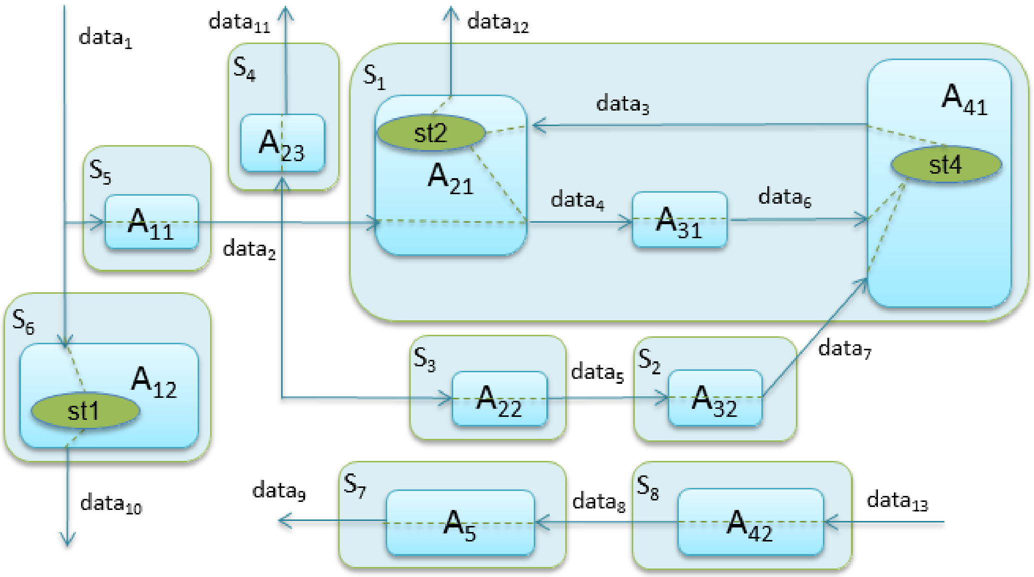


4.11.2 Level L1 system

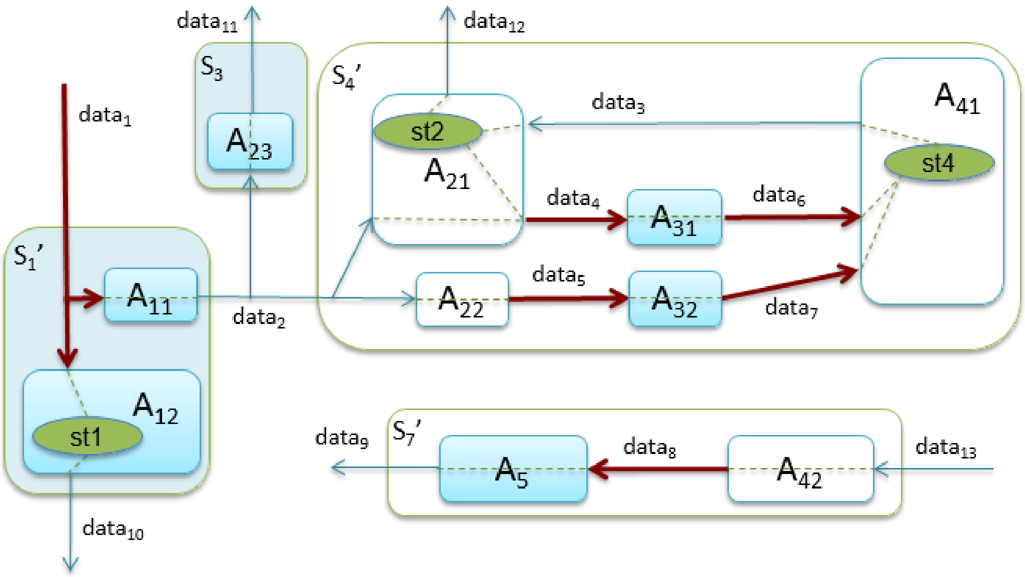


Note: N means when its neither LT, DT or TT

4.11.3 Level L1 detection of strongly connected service



4.11.4 Level L2 system



4.11.5 Level L3 system

5 ISSUES AND RISK REGISTERS

Rating for likelihood and seriousness for each risk			
L	Rated as low	E	Rated as extreme (used for seriousness only)
M	Rated as medium	NA	Not assessed
H	Rated as high		

Grade: Combined effect of likelihood/seriousness					
	Seriousness				
Likelihood		low	medium	high	EXTREME
	low	E	D	C	A
	medium	D	C	B	A
	high	C	B	A	A

Ref:	Description of risk	Likelihood	Seriousness	Grade	Effect on project	Actions
1	Being late at the group meeting	H	E	D	- Lack of commitment - Poor communication	- be on time. - be aware that it could lead to a waste of time
2	Merging each developers programming work	M	H	B	- produces errors - cost extra time	- Exchange the individual work more frequently - Communication
3	Group member suffers from illness	L	H	C	- delays - extra work	- notify the supervisor as well as the group members
4	Misunderstanding of the project	M	M	C	- difficult to continue the weekly task	- Make sure team member someone who understood tries to give an explanation - Try to ask for help from the supervisor
5	Missing each week's timesheet logging	H	L	C	- difficult to figure out each members' contribution	- Project manager tries to remind group member - awareness of group member
6	Delays on allocated tasks	M	H	B	- delays	- try not to procrastinate the work. - manage the time

7	Lack of regular group meeting	L	M	D	- Poor communication - hard to exchange the thoughts on project and catch up	- Arrange weekly basis of group meeting along with supervisor meeting
8	Technical inability for a given feature to be implemented	L	H	C	- delays - Lack of commitment	- ask for help from project members or supervisor

6 TEST CASES (IN PROGRESS)

6.1 TEST CASE 1

Title: fewer services

Case description: The first test case is created in order to see if there are fewer services than usual and then it still goes through the program.

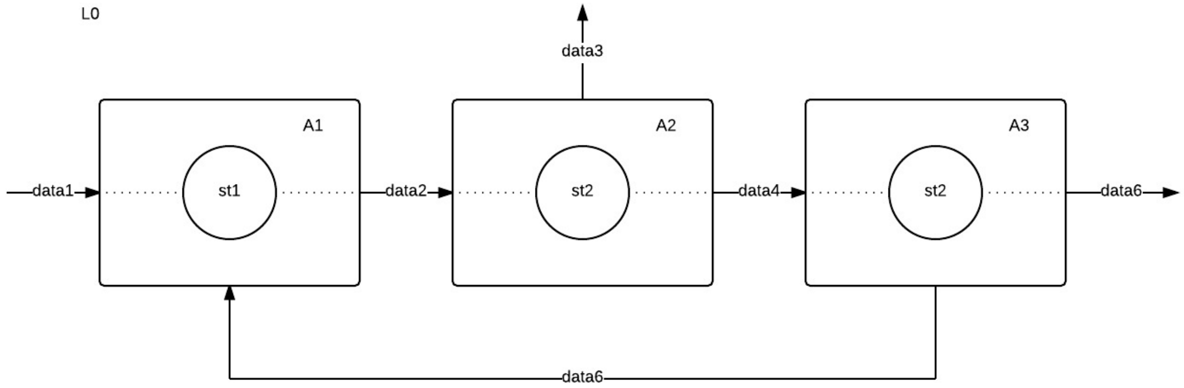
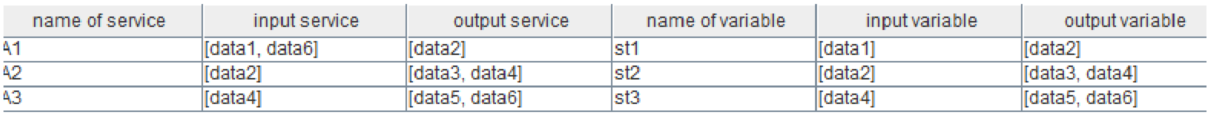


Fig. 1. Dependencies of the services

Generate level 0

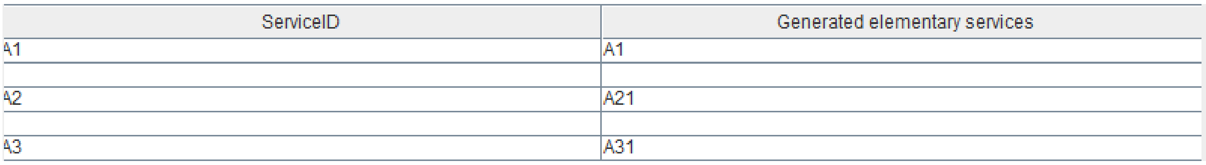


name of service	input service	output service	name of variable	input variable	output variable
A1	[data1, data6]	[data2]	st1	[data1]	[data2]
A2	[data2]	[data3, data4]	st2	[data2]	[data3, data4]
A3	[data4]	[data5, data6]	st3	[data4]	[data5, data6]

Fig. 2. Result of generate level 0

Once its input JSON file is uploaded to the program, it clearly shows the result as shown in the Figure 2. A1 has one output channel which is data2 is depending on the inputs of data1 and data6. A2 has two output channels that are data3 and data4 depending on the local variable st2. And lastly A3 has also data5 and data6 output channels and those are depending on the local variable st#

Check elementary services



ServiceID	Generated elementary services
A1	A1
A2	A21
A3	A31

Fig. 3. Result of check elementary services

It also went through the check elementary services function. Since A1 has only single output channel which is data2 therefore it does not need to be decomposed. On the other hand, A2 has two output channels which are data3 and data4. They are depending on the same local variable which is st2 so all the output channels in A2 are correlated. A3 is the same as A2. It has two output channels: data5, data6. They are also depending on the same local variable which is st3. So the output channels are correlated as well.

Generate level 1

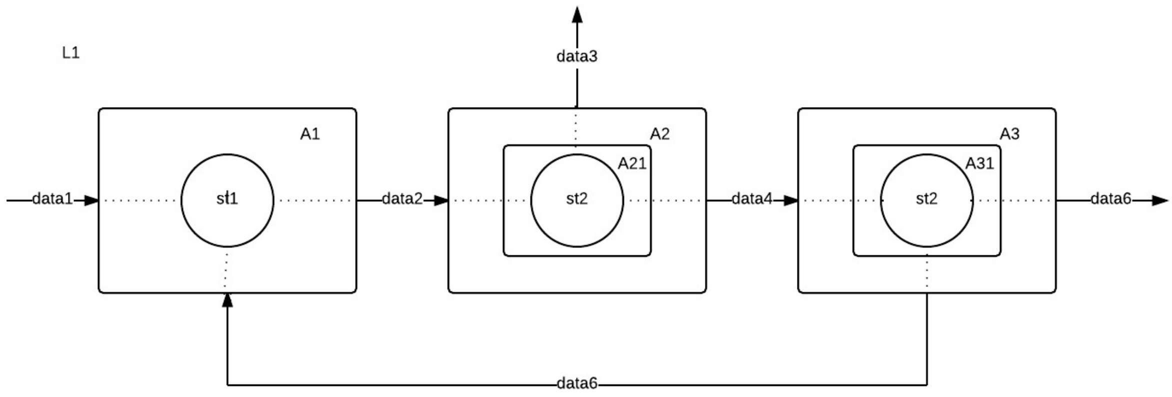
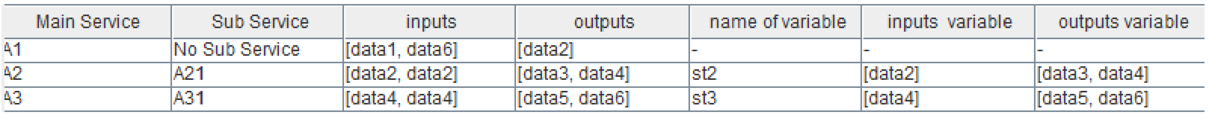


Fig. 4. Services' decomposition



Main Service	Sub Service	inputs	outputs	name of variable	inputs variable	outputs variable
A1	No Sub Service	[data1, data6]	[data2]	-	-	-
A2	A21	[data2, data2]	[data3, data4]	st2	[data2]	[data3, data4]
A3	A31	[data4, data4]	[data5, data6]	st3	[data4]	[data5, data6]

Fig. 5. Result of generate level 1

Fig. 4 is the diagram of decomposed services. As it is explained in the elementary services, A1 does not need to be decomposed because it has a single output channel. And both A2 and A3 have to be decomposed in A21 and A31 respectively since it has two output channels which are correlated to its local variables.

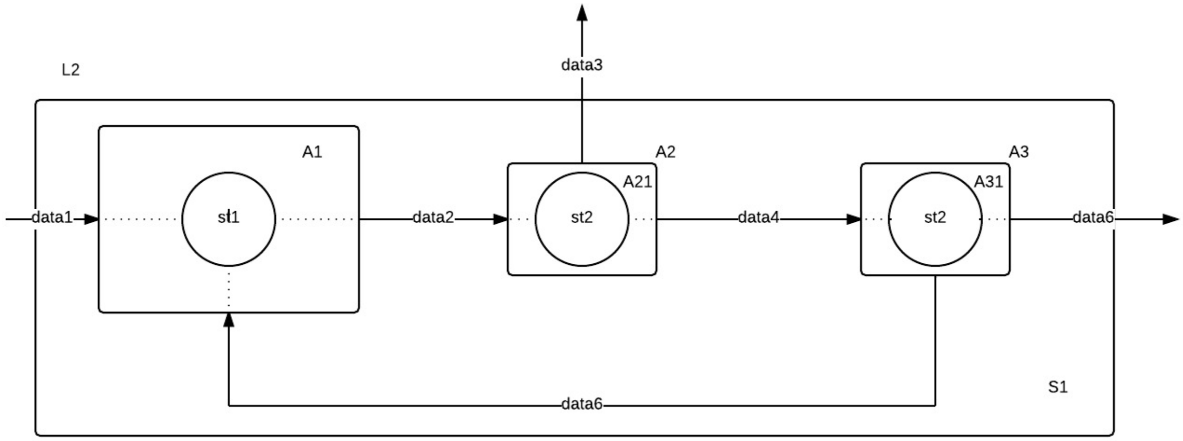
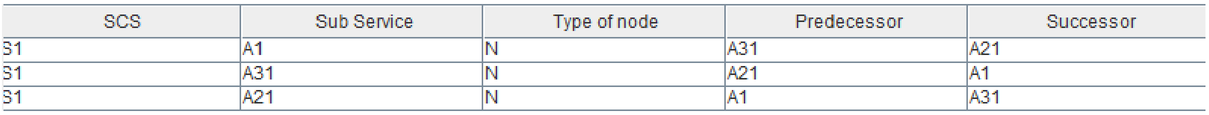
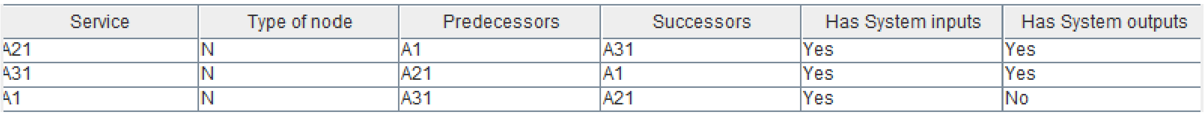


Fig. 6. Diagram of level 2



SCS	Sub Service	Type of node	Predecessor	Successor
S1	A1	N	A31	A21
S1	A31	N	A21	A1
S1	A21	N	A1	A31

Fig. 7. Results of strongly connected service



Service	Type of node	Predecessors	Successors	Has System inputs	Has System outputs
A21	N	A1	A31	Yes	Yes
A31	N	A21	A1	Yes	Yes
A1	N	A31	A21	Yes	No

Fig. 8. Results of generate level 2

As you can see in Fig. 6,7 and 8, each services are covered in S1 because of the node is N which means neither of the services are LT, TT, T and DT. Therefore, it can be grouped as one.

6.2 Test case 2

Title: multiple local variables in service

This test case is for testing when there is two or more local variables and

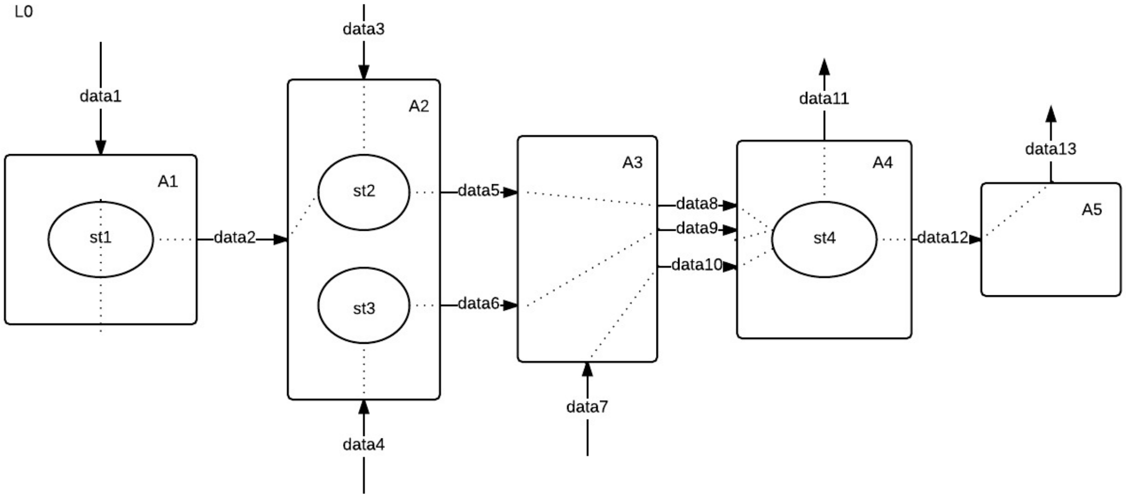
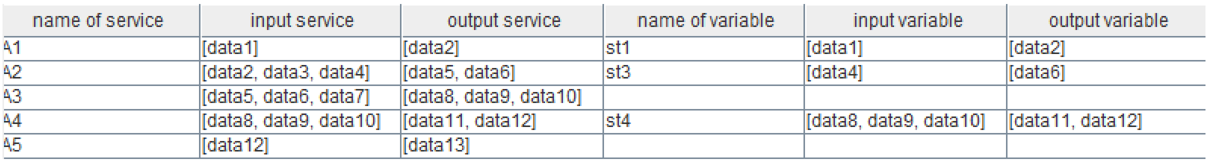


Fig. 9. Dependencies of the services

Generate level 0

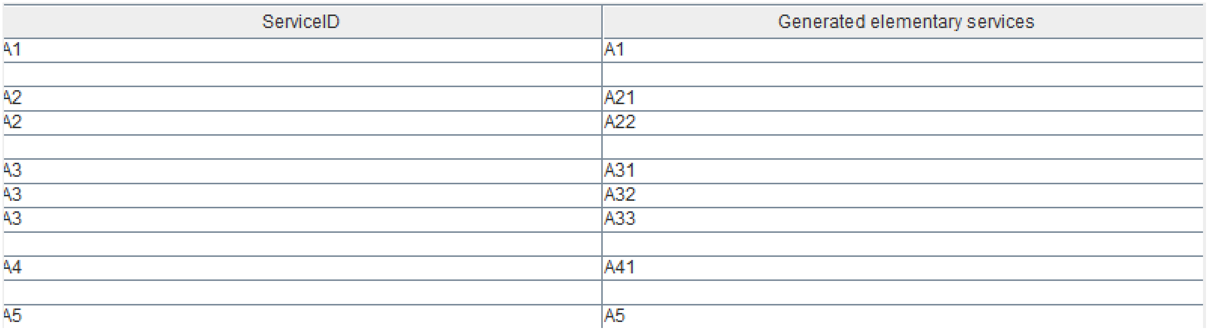


name of service	input service	output service	name of variable	input variable	output variable
A1	[data1]	[data2]	st1	[data1]	[data2]
A2	[data2, data3, data4]	[data5, data6]	st3	[data4]	[data6]
A3	[data5, data6, data7]	[data8, data9, data10]			
A4	[data8, data9, data10]	[data11, data12]	st4	[data8, data9, data10]	[data11, data12]
A5	[data12]	[data13]			

Fig. 10. Result of generate level 0

After uploading the JSON input file to the program, debugging is needed because the result shown above that in the section of the name of variable in A2 has only one local variable. Basically A2 consist of two local variables to check whether it is working in our system. So that had to be fixed.

Check elementary services



ServiceID	Generated elementary services
A1	A1
A2	A21
A2	A22
A3	A31
A3	A32
A3	A33
A4	A41
A5	A5

Fig. 11. Result of check elementary services

Check elementary services for test case 2 has been successful. To be specific, A1 does not need to be decomposed since it has a single output channel. A2 has two output channels data5 and data 6. Data5 is depending on the data2 and data3 via a local variable st2 therefore it should be decomposed in one subservice and data6 is depending on data4 via local variable st3 so it also should be decomposed in one subservice. In A3 each data8, data9 and data10 is depending on data5, data6 and data7 respectively. So it needs to be decomposed into three subservices. A4 has two data output data11 and data12 depending on the same input data8, data9, data10 via one local variable st4 so it also needs to be decomposed. A5 does not need to be decomposed because it has one single output channel and it is depending on data 12.

Generate level 1

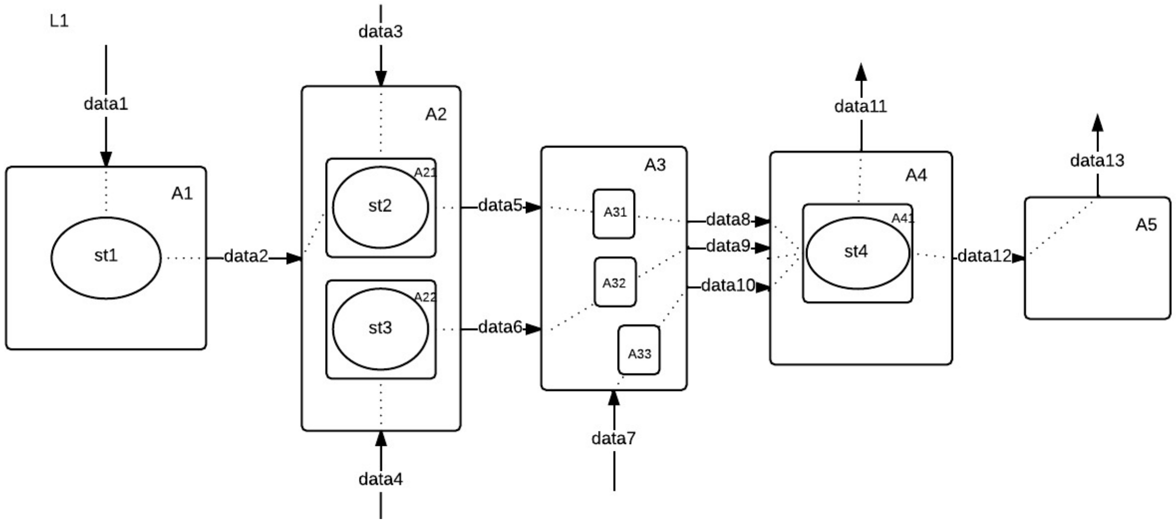
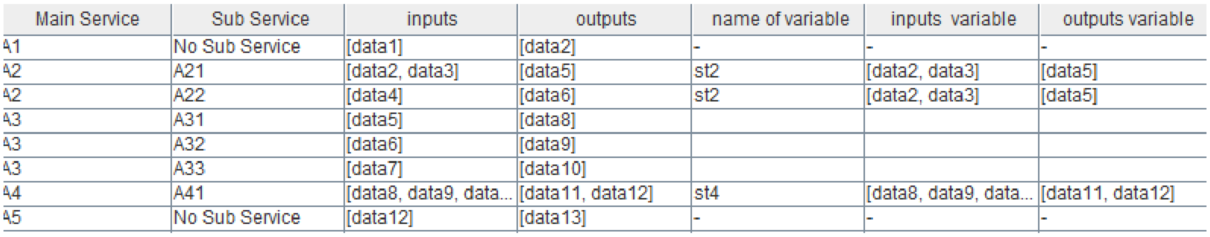


Fig. 12. Services' decomposition



Main Service	Sub Service	inputs	outputs	name of variable	inputs variable	outputs variable
A1	No Sub Service	[data1]	[data2]	-	-	-
A2	A21	[data2, data3]	[data5]	st2	[data2, data3]	[data5]
A2	A22	[data4]	[data6]	st2	[data2, data3]	[data5]
A3	A31	[data5]	[data8]			
A3	A32	[data6]	[data9]			
A3	A33	[data7]	[data10]			
A4	A41	[data8, data9, data...	[data11, data12]	st4	[data8, data9, data...	[data11, data12]
A5	No Sub Service	[data12]	[data13]	-	-	-

Fig. 13. Result of generate level 1

It is explained why it need to be decomposed in elementary services so that it the result of generate level1 pretty much right but it still does not show the local variable which is st3.

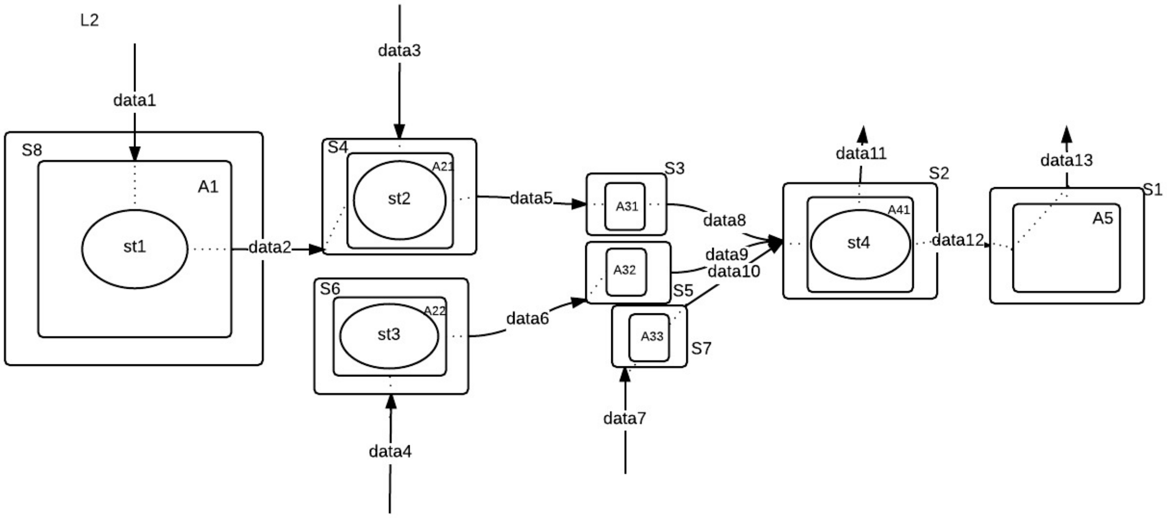
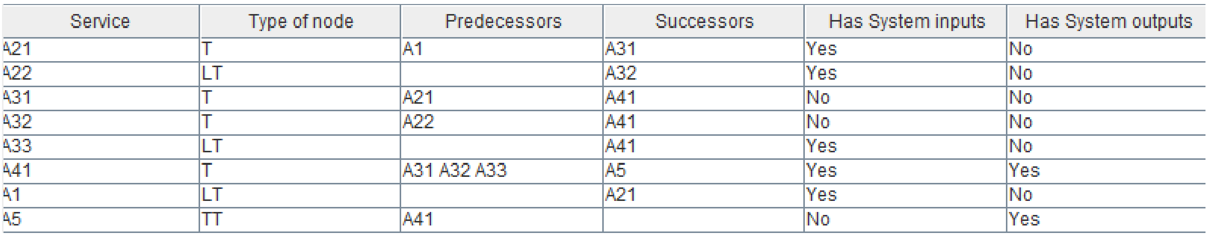
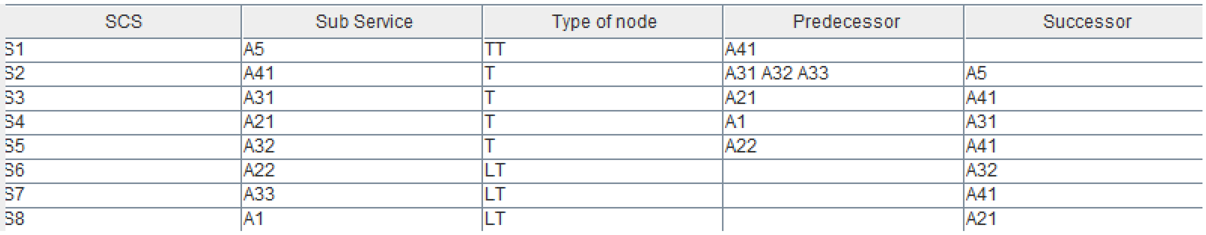


Fig. 14. Diagram of level 2



Service	Type of node	Predecessors	Successors	Has System inputs	Has System outputs
A21	T	A1	A31	Yes	No
A22	LT		A32	Yes	No
A31	T	A21	A41	No	No
A32	T	A22	A41	No	No
A33	LT		A41	Yes	No
A41	T	A31 A32 A33	A5	Yes	Yes
A1	LT		A21	Yes	No
A5	TT	A41		No	Yes

Fig. 7. Results of strongly connected service



SCS	Sub Service	Type of node	Predecessor	Successor
S1	A5	TT	A41	
S2	A41	T	A31 A32 A33	A5
S3	A31	T	A21	A41
S4	A21	T	A1	A31
S5	A32	T	A22	A41
S6	A22	LT		A32
S7	A33	LT		A41
S8	A1	LT		A21

Fig. 8. Results of generate level 2

A1, A22 and A33 is Leading trivial

A21, A31, A32 and A41 is neither leading and terminating trivial SCS

A5 is the only terminating trivial so is is classified as TT.

7 Peer Review (waiting for Yang's peer review)

7.1 Simon's peer review:

Simon Gor
In this project I was the project manager which included tasks such as organising weekly meetings, keeping group members accountable, complete weekly reports and act as the intermediate between our supervisor and group members. In addition the group decided that I would work on the project documentation since my coding abilities were limited. I believe the areas I've performed well in were organising the weekly meetings and act as the intermediate. On the other hand I can improve on keeping each team member accountable since at times some group members weren't performing at a constant standard and lastly do a better job of the documentation.

Zohoor Yahya
Zohoor was one of the designated developers since her coding ability stood out in the group. One thing that I was impressed with was her outstanding work ethic and contribution towards the project. She managed to finish many program features and spend many hours on them. In the meantime Zohoor could work on spending more time with the group since there were times when we needed to ask her questions 1 on 1 and she was unavailable.

Yang Deng
Yang was the other designated developer. He completed many programming subjects and we felt he was best suited for this role. Also as an international student his English was lacking at times which is why we didn't allocate him to do the documentation. One thing Yang did well in was also his work ethic and commitment to the project. He spent many hours programming and getting certain features to work. In the meantime he can work on his communication skills which would have helped him to explain certain software features.

Deok Yeon Kim
Deok was the other designated documenters. Due to his lack in coding skills, this was one of the reasons why we gave him this role. Deok did well in communicating between each group member. He also was committed to every meeting and only missed one. On the other hand Deok's work ethic wasn't great as he couldn't finish some of his assigned tasks.

7.2 ZOHOOR'S PEER REVIEW:

Zohoor Yahya
Zohoor is developer, I have code GUI, Level 1, Strongly connected services, Level 2 and Check property buttons. I could improve the GUI.

Simon Gor
Simon is the project manager, he Keep track of the project, writing down the documentation, arrange weakly meeting and help other members in the project. He could improve keep members contribute more in the project.

Yang Deng
Yang is developer, he code upload Jason file and check elementary button. He did upload Jason file very well. He could improve check elementary and write down comments on the code.

Deok Yeon Kim
Deok is doing the gantchart and write down usecases, He do well in gantchart. He could improve its use cases.

7.3 DEOK'S PEER REVIEW:

Simon Gor

- What contributions did they give to the project?
  As a project manager, He focused on general tasks to keep the project running. For example, logging each weekly timesheet of the members' and manages to represent weekly task to the supervisor. Not only doing that but also, he was mainly in charge of documentation of the project but he was also able to help programmers.

- What tasks did they do well in?
  His role which is project manager, would have been quite a pressure throughout the semester, however he was generally able to help both programming and documentation. He had to be very busy handling a lot of thing related to the project

- What could they have done better and/or improve?
  As he worked as a project manager, when it comes to the time that the group needed to decide something or allocating work to each group member, he could have been more firm and decisive.

Zohoor Yahya

- What contributions did they give to the project?
  Zohoor did contribute a lot in programming work which is pretty significant and she collaborate with Yang regarding coding work and shared each other's thought in order to deliver a better program

- What tasks did they do well in?
  She mainly focused on programming work. With her work, we were able to deliver the program. He focused a lot of time making the program work.
- What could they have done better and/or improve?
  It could have been better if she had more communication with group members. Due to lack of some inefficiency in communication with another programmer, there were slight problem in merging works. Although she was pregnant during the project, she took responsible for her task.

Yang Deng

- What contributions did they give to the project?

Yang has done a lot of programming as well. Starting from gantt chart, he contributed a lot in programming.

- What tasks did they do well in?
  He tried to understand the algorithm and the structure itself. He was pretty diligent and put much effort on programming. And he also tried to help the group with the understanding of the project when another programmer is not around.

- What could they have done better and/or improve?
  In a way of distribution of understanding in programming, there might have been some misunderstanding. It could have been better if there were more chance to talk to him since communication is pretty important in project

  Review
  As I am working on the capstone project, it was pretty good experience for me. From the start I mentioned that I am not actually good at programming and when our group decided each role, they agreed that I can work on documentation. Since the project was software development project so it was important to deliver the software. This did not make me feel comfortable because I could not contribute much on programming especially when our group fell behind. However, despite the obstacles we have experienced during the projects such as misunderstanding of the lead from supervisor, stuck in creating algorithm and absence of group member in a process of project, our group tried to put much effort in order to accomplish what we have planned.

7.4 YANG'S PEER REVIEW :

Yang Deng

- What contributions did they give to the project?
  (1):Design the JSON File and Design Service with get() and set() function in order to transfer JSON File to Service Class.
  (2):Done the Upload L0 button and Generate L0 button.
  (3):Redesign the Check Elementary Service algorithm and implement the function to generate L1.
  (4):Do The Remote Computation part which includes Check High Load Components and Check High Performance Components.

- What tasks did they do well in?
  (1).Using Gson library to manipulate JSON file and make them more efficient when we need transfer JSON file to Class we need.
- What could they have done better and/or improve?
  (1).I can reduce all the warnings.
  (2).Remove some redundant functions and keep them short and simple.
  (3).Should doing better comments for the others to get understand easier.

Simon

- What contributions did they give to the project?
  He is the manager of our project and done well in the job of documentation.
- What tasks did they do well in?
  Simon done well in the job of communication, documentation especially about the whole project progress control.
- What could they have done better and/or improve?
  Simon can improve his art of management, sometime he need be more strict.

Zohoor

- What contributions did they give to the project?
  Zohoor is the developer who designed the GUI, old Level 1 which need redesign, Strongly connected services, Level 2 and Check property buttons.
- What tasks did they do well in?
  Zohoor done well in the GUI design and help us rapidly arriving, the Strongly Connected Service is the hardest part of our project ,she handled it.
- What could they have done better and/or improve?
  The Algorithm of Strongly Connected Service can be made by herself than can be easier for other member to get understand.

Kevin

- What contributions did they give to the project?
  Kevin done the job of part of documentation and the test case.
- What tasks did they do well in?
  His test case helped the developer Yang to detect two bugs in the code.
- What could they have done better and/or improve?
  Kevin can do more cases in his test case design.

8 TECHNICAL SOLUTION DESIGN DOCUMENT (IN PROGRESS)

8.1 WHY WE CHOSE JAVA OVER PYTHON

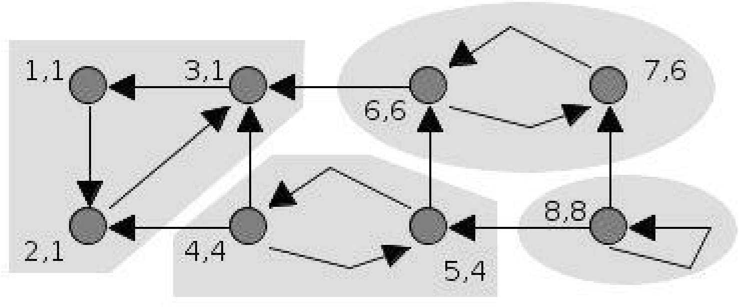
When we came together as a group and had our first meeting, one of the important questions asked was which programming language to use. There were two factors which made us choose Java over Python. The first factor was Yang and Zohoor (our developers) have both worked extensively with Java which lead to them agreeing that they were more comfortable with coding Java. The second and last factor was when came up with a list of pros and cons for both languages. In the end we saw more pros for Java and decided this would be the best language for this project.

8.2 WHY WE CHOSE TARJAN OVER ONE WAY CATCH THEM YOUNG (OWCTY)

When our group reached the Strongly Connected Service feature in the project, we came across the OWCTY algorithm which is used to eliminate LT and TT services. At the time we did some research on the algorithm and found it difficult to find many publications. Many of our search results ended up linking us to sample articles which costs money and there were no free publications available. Even Wikipedia didn't have any record of this algorithm. This was the main factor and reason why we started looking for other available algorithms. After a couple of days Zohoor came across another algorithm called Tarjan. This algorithm created by Robert Tarjan is to help find the strongly connected components of a graph. This was just what we needed and thus why we implemented it.

8.2.1 What is Tarjan algorithm?

The algorithm takes a directed graph as input, and produces a partition of the graph's vertices into the graph's strongly connected components. Each vertex of the graph appears in exactly one of the strongly connected components. Any vertex that is not on a directed cycle forms a strongly connected component all by itself. An example of this is when a vertex whose in-degree or out-degree is 0, or any vertex of an acyclic graph.



A representation of an acyclic graph in which Tarjan algorithm has been implemented.

For more in-depth detail on Tarjan, visit the link below:
https://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm

For an animated GIF of how Tarjan algorithm works, visit the link below:
https://upload.wikimedia.org/wikipedia/commons/6/60/Tarjan%27s_Algorithm_Animation.gif

8.3 WHY WE PICK JSON AS AN INPUT FILE

When it came to selecting an input file for our project, there were two file format standouts. The first one being XML and the other JSON. Both languages were applicable to our project but we could only use one. There were 3 underlying factors as to why we picked JSON over XML. The first factor was the flexibility and usability of JSON. Since JSON is a lightweight data-interchange format, it was meant to be easy for humans to write and read. As a group we also wrote two input files, both in XML and JSON and found JSON much easier to write and modify. The second factor was the available Java libraries that helped us achieve what we were after. When we did our research on JSON java libraries, we came across GSON. This google library had all the tools and functions we needed and since it has a big following on GitHub, it was perfect for our project. The third and last factor was Yang's experience with JSON. In year 2 of his course he studied a course that involved JSON files and he knew how to create them. This was a big bonus and saved our group a lot of time.

9 LEARNING OUTCOMES

9.1 ENABLING KNOWLEDGE

Information technology and computer science students have completed a variety of subjects in order to reach the capstone project. These subjects include first year programming which focused on languages such as Java and Python. In addition, year 2 gave us further opportunities to complete advanced programming subjects that built on our Java and C# knowledge. These programming subjects gave us a range of knowledge from solving simple tasks to more complicated problem solving puzzles. Therefore in this project our developers were able to utilise and apply these skills to develop and code the project's software. Apart from programming skills, subjects like Software Engineering Project Management taught us how to write documentation (project charter etc) and how to apply our knowledge of project methodologies and use it to guide each member in order to have a successful project. Without this subject our group would've struggled to manage, plan and execute the project requirements.

On top of applying our skills from previous year subjects, we also developed and learnt many new valuable skills/techniques throughout this project. An example of a technique we learnt was how to develop the software so that it can upload and read the contents of a JSON file. At the beginning this was particularly challenging since none of our group members were experienced in using the JSON file format. This led us to research and utilized online forums/programming communities like Stack Overflow to successfully create the Upload L0 feature. Furthermore we learnt a valuable skill in critically analyzing papers. This skill is only briefly brought up in some subjects but not too the extent of analyzing a paper and creating a software based off it. We learnt how to use a systematic method that allowed us to properly analysis the paper in order to implement each feature.

9.2 CRITICAL ANALYSIS

In a programming project like this one, the agile project management methodology suits it perfectly. Continual updates and iteration adds more value to the next until you have the final product. An example is when you're developing a new product. You are bound to have bugs or missing features that affect the output of the software. Our developers faced many situations like this with Check elementary service and Find strongly connected service since the algorithm behind these features didn't account for every scenario. But through using the agile project management methodology they were able to update these features frequently and fix any issues that emerged until they worked as planned.

A big part of this project was to critically analyse the different sections in the paper which corresponded to a feature. When it came to analysing Strongly connected service, it was recommended that we used the OWCTY (One way catch them young) algorithm. As mention in the System solution section, this algorithm was very unique and the resources online were very limited. Because of this we weren't able do a critical analysis on this algorithm and had no choice but to use the Tarjan algorithm which is well publicised on the internet.

9.3 PROBLEM SOLVING

In every project problems and issues can occur and surprise everyone at any stage. Whether it occurs at the beginning or close to the end of the project, it can take a significant effect on your group members and the project outcome. The first major problem faced by our group was underestimating the duration it takes to complete each feature. We initially thought that most features would take only 1 week to finish but unfortunately this wasn't the case. An example of this is when we were creating the Upload L0 feature. On paper it sounds like an easy task (upload file and read file etc.) but the decision to read a JSON file brought many issues. There were many Java libraries which were capable of reading the file and in the end we tested 2-3 libraries. Each library took at least 2-3 days to implement with success on the third library.
The second major problem we faced was not a technical one but rather a circumstance. One of our group members had health issues and had to take a two week break. This was a big surprise to our group since we didn't see it coming and pushed our software development further behind. The only solution in this case was to continue our individual assigned tasks, communicate with our supervisor and apply for an extension.

9.4 COMMUNICATION

-Had some issues with our supervisor. E.g. misunderstandings and didn't ask enough

When it comes to the importance of communication in a project, good communication allows teams/groups to be on top of their work, reduce the risk of conflicts, produce a quality final product and overall have a successful project. In this project we decided as a group to communicate using three different methods. The first method was to communicate outside of university hours through the use of social media. In most cases groups will create a Facebook group or use the Facebook chat service to communicate to each team member but in our case not every group member used Facebook. In the end we used a mobile phone application called WhatsApp since this application was the common denominator and used by all of our group members. The second communication method was to hold weekly meetings which is a part of our agile methodology. This weekly meeting was essential to update each group member on how they performed during the week and the tasks they've completed. Lastly at the end of the meeting we would talk about the tasks we would undertake before next week's meeting. The third and last communication method was between our group members and supervisor. Whenever someone found a particular task difficult or hard to understand, we would email our question/requests to our supervisor as a way to get help. Email was the only option in this case since our supervisor's schedule was busy during the week (lectures, other group project responsibilities etc) and it allowed her to reply at her own pace.

Throughout this project our communication had many positives as well as some negatives. The positives were our weekly meetings and communication through WhatsApp. Every group member made the weekly meeting as a priority as well as being prepared for it. This resulted in each meeting being efficient and effective which helped this project tremendously. Meanwhile our communication through WhatsApp was also a standout. All of our group updates and reminders would be communicated through this application which helped organise weekly meetings and general catch-ups. The other positive with the group communicating through WhatsApp is the liveliness. At least 5 days out of the week we would communicate and help each other when there was a request from a member. This helped us to bond better and work as a team. Lastly, as a group we could work on our communication with our supervisor. At times when we are listening to instructions and some instructions were hard to understand, we lacked the initiative to ask questions in order to clearly understand the required tasks. This in turn led to misunderstandings and uncompleted tasks.

9.5 Teamwork

At the beginning of the project, members are allocated with one of three different roles such as project manager, documenter and developer. The project manager's role consisted of traditional tasks such as weekly reports (agile methodology), documentation, main communicator between supervisor and group, project planning and project governance. The documenter had the tasks of writing both technical and non-technical documents. An example of a technical document is the system design and a non-technical document is the project charter. Finally the developers role is to construct and code the software as well as conduct debugging in order to have a functional program. Although these roles were allocated to group members, it was expected that each member would give a helping hand to a group member who hasn't been assigned the same role (E.g. Documenter helping the developer with his coding). This was one of the ways we worked as a team.

When it came to the differences and circumstance faced by the group, one particular circumstance stood out that required each group member to unite and work as team. In this particular circumstance a group member (developer) was facing some health related issues and had to take 2 weeks off in order to recover and get their health back on track. In these 2 weeks our remaining group members took on the initiative to continue the unwell member's work and attempted to add more content. This was particularly challenging since the code was written in their style but through teamwork and management we were able to add a couple of features.