The Software Engineering Group Project 2017

Prof. Martin Neil & Dr. Mustafa Bozkurt

Group Project Objectives

- Simulate industrial practice
- Work as a member of a project team
- Create and follow a project plan with defined deliverables
- Gain experience of using techniques taught
- Complete the project on time

The Group Project Documentation

- Two key documents:
  - Problem Definition document describing Functional and Non functional requirements of system to be developed
  - Project Guidelines document specifies how the project is structured, the deliverables expected and the assessment scheme applied
- READ THESE DOCUMENTS EVERY WEEK

Organisation

- Groups of 5 students
- All groups work independently of others
- Membership is not negotiable
- Groups identified by pre-assigned identifiers
- Each group must appoint its own project manager

Attendance and Meetings

- Weekly Lab session in ITL (half hour with TA)
- Weekly checkpoint meeting
- Additional group meetings, time and location at discretion of group

Attendance in the ITL is mandatory and records will be kept

Communications and Support

- TAs are allocated to each and every group
- Use the QMPlus module page – it is the main source of information
- Set web page forum to ‘watch’ and check email
- Post *all* (non-personal) queries to web page forum
- Use email for internal messages amongst group members
- Use email to the Prof. Neil or Dr Bozhurt for *major* issues specific to your group. *You must include your group letter in the header and your own real name in the body in any email communication*

Checkpoint Meeting Agenda

- Groups must hold weekly ‘checkpoint’ meetings
- Standard agenda
- Need only last 15 minutes
- Project manager is chair
- You will lose marks for not submitting form as coursework

A record of promises made, then broken or fulfilled

Checkpoint minutes - Register

Checkpoint Meeting Minutes

Group Identifier: [ ]

Date of Meeting: [DD] [MM] [YYYY]

Attendance Register:

Name	Present

If present mark with (X), approved absence (A), otherwise leave blank

Checkpoint minutes – Review past week

Progress review of last week

Name	Task Description	Task Status
Person 1		
Person 1		
Person 1		
Person 2		
Person 2		
Person 2		
Person 3		
Person 3		
Person 3		
Person 4		
Person 4		
Person 4		
Person 5		
Person 5		
Person 5		
Person 6		
Person 6		
Person 6		

Max three rows per group member. Task status: (C) completed (O) ongoing (X) no report

Checkpoint minutes – Plan next week

Plan for next week

Name	Task Description	Task Status
Person 1		
Person 1		
Person 1		
Person 2		
Person 2		
Person 2		
Person 3		
Person 3		
Person 3		
Person 4		
Person 4		
Person 4		
Person 5		
Person 5		
Person 5		
Person 6		
Person 6		
Person 6		

Max three rows per group member. Task status: (N) new (O) ongoing (X) no report

Checkpoint minutes - Risks

Other Issues and Risks

Project Roles

- EVERYONE will be expected to program java code, without exception
- EVERY student will be allocated a module of the system to be developed
- ALL students must apply software engineering methodology and design principles throughout
- DO NOT assume you can do your module in ISOLATION. Each module has dependencies with the others so that delivery of an integrated system cannot be avoided
- ALL students will act as analyst, designer, programmer and tester

Garage Management Information System (GM-SIS) - Problem Definition

- Comprised from these modules:
  - Authentication
  - Customer account
  - Vehicle records
  - Diagnostic and Repair bookings
  - Parts record
  - Specialist Repairs
  - Scheduled maintenance bookings

Problem Definition

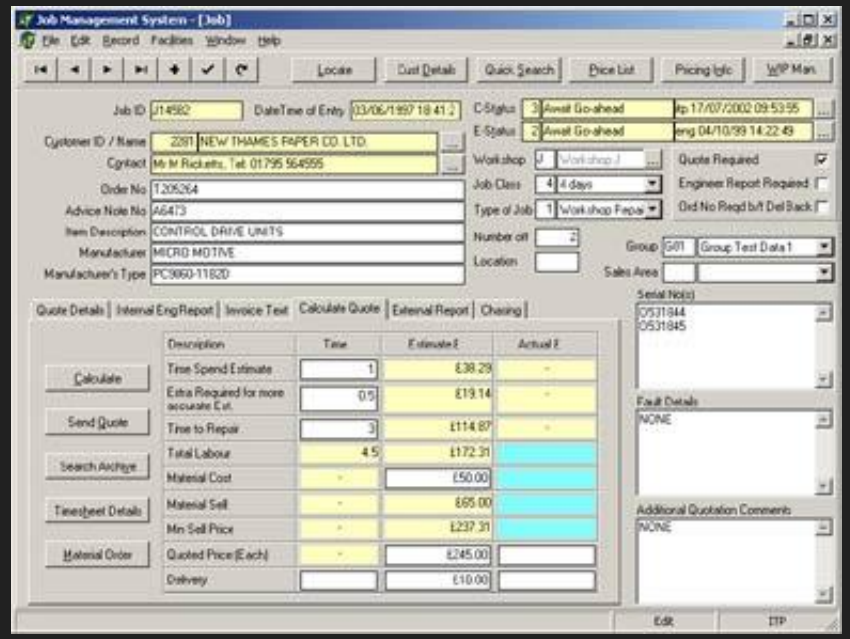
Each module will need to be developed separately and meet the following functional requirements:

- Present a graphical user interface to the user to allow them to carry out all operations.
- Allow records and data to be added/edited and deleted.
- Correctly implement the specific data requirements associated with the module.
- Save and retrieve records to/from a database table(s).
- Execute specific operations on the data associated with the module.
- Provide a method of user authentication.

Databases and other stuff

- There is a requirement for the system to use a database for storage and retrieval of all data
- Not all students have used a database before, but almost everyone is doing the database module
- You can wait until you have accumulated the knowledge needed to build the database before committing to it
- In any case your UML object model should be easily translatable into an ERD for a RDMS
- Doing this project WITHOUT a database would be a major headache and would involve MUCH more work, so this is saving you a LOT of time
- Anyway, why recreate the wheel? Same goes for code generators, UI designers etc. – use what you want so long as it is Java compliant and documented in your notes

Garage Management Information System (GM-SIS)



Module Priority Order

(ALL) Authentication

(A) Customer Account

(A) Vehicle record

(A) Diagnosis and Repair Bookings

(A) Parts

(B) Specialist Repairs

(C) Scheduled Maintenance bookings

- Do A modules only if group has 4 members
- Do A and B if group has 5 members
- Do A, B and C if group has 6 members

Group Repository

- Mandatory use, by every group and member, of configuration management system called “Git”
- Manage ALL documentation and revision history via a Git repository
- Allows you to track and back track on every change made and will ensure easy integration of code
- All documentation to be controlled under Git too, including project management materials

Git folder setup

- Master
  - Release
  - Documents
  - Working
  - Checkpoint
- Scratch
  - Sub folder for group member 1
  - Sub folder for group member 2
  - Sub folder for group member …

Group Project Assessment

Deliverable	Week	Deliverable date	Delivery mechanism	Weighted marks per group member
Checkpoint meeting minutes	2-11	Each Friday midnight 23:59:59	Coursework submission system	10
Project repository submission and revision	2-12	Continuously	All group materials, source code, supporting libraries and databases, documents, test cases and checkpoint minutes	10
Release (of integrated system)	12	Final deadline Friday 31 March 2017 23:59:59	Coursework submission system	60
Test and Contribution report	12	Final deadline Friday 31 March 2017 23:59:59	Coursework submission system	20
Total				100

Individual Vs Group Contributions

- Hard for poorly performing members to ‘hide’
- Final mark will reflect individual contribution – you will be asked to rank the contribution of other group members
- Individual reports submitted at end of the semester
- Marks will also be adjusted for individual contribution based on checkpoint minutes, lab register and marks for modules

Group Dynamics

- Not everyone can be leader but be careful of democracy
- Do not let members take a back seat
- Do not be fooled into thinking you can do it on your own

4 Stages of Group Formation

1. Forming
2. Storming
3. Norming
4. Performing

- If you are at stage 2 after four weeks you *will* fail.

Some Caricatures

Beware these personality types……

The Prima Donna

The Regurgitator

The Bully Businessman

The Great Pretender

The Programmer from Jurassic Park

Disputes

- You will have disputes! Think hard in advance about how you might resolve them
- Get teaching staff involved as LAST resort
- Remember to treat people in the way you would like to be treated
- DO NOT
  - Ignore poor performance, or
  - Fail to record it on the checkpoint minutes
- DO
  - Confront the issue
  - Assume there is a reasonable explanation

Treat it like you would a zombie attack…

Your survival will depend on good planning, teamwork and strong execution!

Technology

- Netbeans IDE
- Database: SQLite
  - SQLite JDBC Driver
  - https://bitbucket.org/xerial/sqlite-jdbc/overview
- QMUL GitHub Enterprise
- Operating System
  - Windows
  - Linux

First Lab

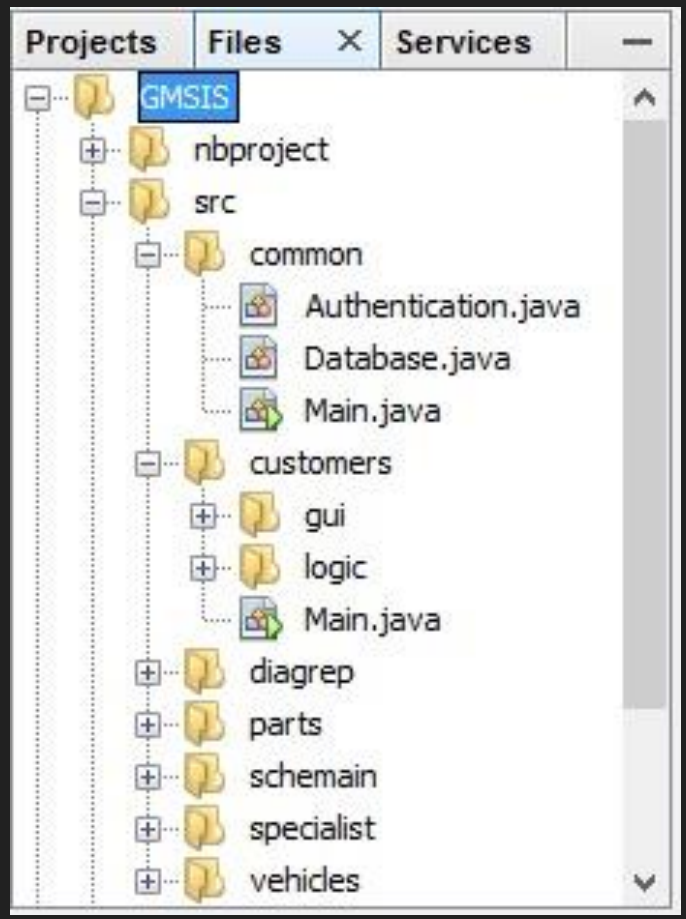
- Labs start from week 2, January 19th
- Meet your TA
- Set up Git
  - How to log in (link)
  - How to access organisation account (link)
  - ProGit covers both basic and advanced usage
- Set up Netbeans project
- Learn your deadlines

Tips

- It is easier to develop an integrated project than integrate completely separate pieces later
- Identify common features used by all project components
- Identify data shared between components (i.e. coupling)
- Work closely with team mates who you have common data with
- Concentrate on implementing requirements from project definition rather than bonus features or fancy looks

Recommended NB Project Structure

- Common package for shared features
- Own package for each component
- Separate GUI from application logic
- Own main method in each component
- Example to the right →



Git reminder (.gitignore file)

- Not all files have to go into the git repository e.g. db, libraries, configuration and build files.
- Make use of git ignore functionality to make sure you don't overwrite each others conf and build files.

Release

- Jar files and libraries
- Packaged
- Configuration and connections automated
- Machine and platform agnostic
- Zipped submission – approved by team
- Do a dry run of the packaging and test this process before submission

Test and Contribution Report - Outline

- Describe your design
- UML diagrams mandatory
- Assess contributions from team members
  - Confidential rankings E, M, B, F
  - This will affect marks awarded
  - But we will look for supporting evidence (Git, Checkpoints)

Test and Contribution Report – Test Plan

- You deliver test plan and test cases with test results
- This will be used by us to test your system
- Failure to do this could be fatal
- Makes sense to have your system and testing completed and documented well before deadline

Easy and Boring Ways to Lose Marks

You may lose marks if you or your group:

- Fail to adhere to the Seating Plan
- Occupy space allocated to another group
- Miss meeting minute submissions
- Fail to fill in meeting minutes template properly
- Fail to use Git regularly (more than once per week)

Q&A

Questions?