A picture containing text

Description automatically generated

School of Computer Science

UG Capstone Projects

First Group Report Template

# Reports Requirements

## Goals of the first report

* Draft for final report
* Feedback to inform
  + group on how it is doing
  + individuals on how they are doing
  + direction of project from here on
  + guidance on refactoring
* Inform us of how you are doing
* Ensures appropriate scope for project
* Ensures appropriate allocation of work for individual

## Some Pragmatics

Be aware that the report is an important part of the course. It may not feel as important as designing the system and tests as well as writing code, but it is critical for communicating what you have done.

It is also a collaborative endeavour that should run through the semester. To make this easier, you must **make** use of an authoring medium that supports collaborative writing of documents.

You must write the report as a document in your wiki. This gives you excellent support for group authoring.

## Format of the first report

1. Cover page (Template Cover Page + Official Group Assignment Coversheet)
2. Executive summary (Single Page)
3. Table of Contents
4. Main chapters/Sections (Include section headings and subheading)
5. References
6. Appendices, including each member's brief **one** page summary of individual report that highlights the **most important contribution for each week as a member**.

**Note:** All links throughout the documents should be [hyperlinked](https://canvas.sydney.edu.au/courses/27761/assignments/250501) and cross-referenced rather than providing the complete URL.

Any broken/unreachable link will not be considered as evidence. Any evidence based on blurry image will not be considered. All tables, figures, diagrams, images should be numbered and captioned.

### Cover page

(Please use the **template below** as the **first page** of your project report and make Standard university cover page for group work: “**ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT”** (declaring member’s contributions and their signatures) as the **second page**)



**Project Name**

**(e.g., CP1 - Autonomous Car Using Neural Networks and Computer Vision)**

**Client name**

**Group name (e.g., COMP3888\_T17\_03\_Group3)**

**Tutor / supervisor name**

**Group members details (indicate if any member is doing multiple units)**

**Submission Date**

### Executive summary (Single Page – 3rd Page of your report)

* Project title
* Group members
* Project aims/goals
* High level description of what your system does/major achievements, written for the busy executive client
* Acknowledgements (as appropriate)

### Table of Contents

This is self-evident.

It will be important to format the report so that each section can be easily found, for example, using coloured tabs at the start of each section, with the contents written/printed on it. This is especially important in the appendices which the markers will need to consult as they read other parts.

### Introduction (1 page)

* + Problem statement
  + The overall project vision, goal, purpose or objective
  + What the project will achieve
  + The key stakeholders, what do they do, and how they interact
  + Identification of resources and risks involved in the project

**Note:** Using XP means that you should always have a set of user stories and the associated priorities from the client and, potentially a risk rating. All these details will be available, with their histories (such as where the client has altered the priority of a story) on the Wiki. At any stage, those that have been allocated to be done (as milestones) will have associated engineering tasks (as tickets). The introduction documents just the user stories that have been completed and are expressed in their final refined form. (Markers can and will use the Bitbucket site to see their history.)

### Overview of system from user view (user stories)

#### Overview

* In first report, it could be the version for current draft/prototype as an indication of what you plan to achieve.
  + - 2.1.1 User
    - 2.1.2 Administrator
    - 2.1.3 Other users....
* For each class of user, provide a small selection of screen shots which show the core functionality of the system via user scenario walkthroughs.

### Evaluation

**Note:** It should show structure and current drafts of requirements and acceptance tests

#### Overview

* Clearly described and relevant, demonstrate how they contribute to the client’s needs, testable, are bite-sized, estimable, have clients’ rationale and have acceptance criteria
* Covers both functional and non-functional requirements expressed in user stories
* Can include high level designs (e.g., mock-ups, UIs)

#### Details of Test

* Main testing techniques involved including unit tests and acceptance tests, with critique of what has been achieved;
* Quality of testing employed including;
  + - Use of relevant testing types (e.g., unit tests, regression testing, integration, system, usability and/or acceptance testing)
    - Evidence of test-first approach and use of tools and processes for systematic/automated testing
    - Use of relevant testing techniques (e.g. Equivalence partitioning, boundary analysis) to design test cases
    - Demonstrate execution of designed test cases and corresponding results (e.g., bugs, bug fixes)
    - Demonstrate enough test coverage; test cases consistent with user stories and cover different aspects of all user stories and why is it enough for the scope of the project
* For acceptance test: ideally provide a table, rows are the requirements and columns are the acceptance tests. List the details of acceptance tests after the table.
* Refer to a summary of the details of the acceptance tests in
* the appendix - reader should be easily able to find this both using links and helpful tags etc.

#### Conclusions

* Summarise what the tests demonstrate, their limitations and the
* Limitations of the system

The full set of actual acceptance tests should be lodged in the repository. The appendix will give details of how to run acceptance tests. For usability testing, the full details will be in the Wiki.

All relevant supporting material in appendices should be cross-referenced, cited as appropriate. (The marker will only read appendices if directed to do so in text body. Evidence should be hyperlinked to appendices and or bitbucket)

Make sure things are consistent and that there is evidence for all claims you make.

### System structure overview This may include:

**Note:** It should be the proposed system architecture – VERY IMPORTANT for feedback.

Single page overview of system structure

This will refer the reader to relevant Appendices if necessary. All aspects of the description will make use of appropriate representations you have met in other units of study. For example, ER diagrams, UML ... as appropriate.

Indicate which components/parts are developed by the group, and which ones are used or need to integrate with.

Remember to refer the reader to relevant appendices.

### Tools to build system

* It should include:
  + - What tools, and **why used, decisions, rationale**
    - Languages, frameworks used for each part
    - Database tools
    - Source code management
    - Other tools that might be specific to the project nature or requirements

### Information search/research and discipline knowledge use and application

This includes concise description of the use and application of your discipline knowledge (e.g., software engineering/development, computer science, data science, science/technology) and search for existing solutions/tools in solving complex problems. This includes tools/systems/technologies/frameworks/ algorithms used to build the system or solve the problems faced during the development

This section should be written at a level that makes sense to a person unfamiliar with the project but with some technical background. In particular, it should make sense to your client and should be understandable be a fellow third year Computer Science/Software Engineering/Science student who had no involvement in your project area. Some elements that you might have consulted and could report include:

* + Software, system, services, APIs, tools
  + Algorithms, data and special equipment
  + Reports and evaluations of software
  + Literature found on the web
  + Literature found in recognised digital libraries for computing
  + professionals.

You are expected to use proper referencing and citations. (See the last section)

### Group processes, reflections and conclusions

Critical assessment of project and processes by group:

* Challenge / Risk analysis
* Limitations in terms of functionality
* Limitations in terms of structure, design, implementation
* Primary strengths
* Programming practices
  + - \* Reflections on Extreme Programming
    - \* Ways of applying version control, issue tracking, coding styles
    - \* Group aspects as well as product and processes

In first report, you can provide a draft based upon planning and experience to date on \* items.

### Individual contributions (individual reports), work split, –

This section contains the work split. Be sure to provide pointers to Wiki/Repository details and report page as relevant.

* + Overview of the system structure and other main elements of the work (including non-technical such as report writing) with names of group members who did each part. Please organise this so it is easy for the marker to see just what each person did. For example, provide a table of tasks, with a column for each person and a tick against the cell showing who did each part. The tick should be hyperlinked to your repo as an evidence. **Be consistent (e.g. the same task should not be attributed to two people and if they did work together, indicate so.)** List all the technical elements together and group other aspects logically. All technical/design elements should be linked to the repository.

Table of contributions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Note that it is **YOUR** responsibility to ensure that the marker can easily find the contributions that you have made. If the marker cannot find relevant evidence on each aspects of the assessment, they will, regretfully but certainly, give no marks for that aspect.

Ideally colour coding to make it very easy for the marker to find the parts.

### Appendices: (No Page Limit)

Start with a cover sheet exactly as for main report. Add details of how to run the system - as applicable:

* + URL for all necessary instructions for the marker to run the system and access code etc
  + Note that all MUST run on the School of IT machines so urls etc must be for School of IT machines.
  + Note that the url should take the marker to clearly presented and complete instructions with details of how to run the system. Make sure you include relevant passwords.
  + Note that all repositories will be frozen at the end of the last lab session. Markers will be able to test code. Note that it is your responsibility to provide all information needed for the marker(s) to test all aspects of your system, and to verify all aspects of the report

## A1. User stories - should be solid though not final for Report 1

These should be available at the Wiki with the full record of their development available to the marker. This section should provide the final version of them to make the report stand alone. In addition, provide the relevant url for the Wiki.

## A2. Research, studies of similar systems - should be substantial for Report 1.

This should have also been built up over a period and that development should be captured by the submissions to the Wiki/Repository. This section should briefly summarise what your team learnt. Provide details of where this is so that the marker will be sure to be able to find and assess it.

## A3. Unit testing summary - should be substantial for Report 1 for work so far.

This complements Section 3 with additional details as necessary for that section. Full history will be in Wiki/Repository. Provide details of where this is so that the marker will be sure to be able to find and assess it.

## A4. Acceptance Testing - should also be substantial for Report 1, for work so far

Similar to unit tests.

## A5. Usability Testing and other - early usability results should be here for Report 1. performance testing as appropriate.

Similar to unit tests.

## A6. Documentation for main modules of the system (automatic generation) - also should be current for Report 1.

Similar to unit tests.

## A7. Other appendices as appropriate

One helpful thing at this point is a summary of the relevant additional information available at your Wiki/Repository. Anything that you have done and is not reported above should be here. Examples include pointers to the details of client interaction and group meeting minutes.

## 

### References:

Example of a book:

citation form in text [Pfleeger 2001] reference at end:

Pfleiger, S L, Software Engineering, Theory and Practice, Prentice Hall (Second edition) 2001.

Example of a manual:

citation form in text [ParcPlace 1991]

ParcPlace, User's guide for Objectworks/Smalltalk, Release 4 (1991), ParcPlace Systems, Mountain View, USA.

Example of an online resource: citation form in text [Perl 2001]

Perl Style Guide, [http://www.perl.org/press/style-guide.html,](http://www.perl.org/press/style-guide.html) visited October 2004