



**THE UNIVERSITY OF THE WEST INDIES**

**DEPARTMENT OF COMPUTING AND INFORMATION TECHNOLOGY**

**COURSE MANUAL FOR  
FINAL YEAR  
UNDERGRADUATE PROJECTS  
INFO 3604**

**V 1.3 November 22, 2018**

# Table of Contents

<b>1. Purpose of the Project Course</b>	<b>4</b>
1.1. Aims and Goals	
1.2. Course Delivery	
1.3. Learning Outcomes and Objectives	
<b>2. Prerequisite Courses</b>	<b>5</b>
<b>3. Project Descriptions</b>	<b>5</b>
3.1. Project Scope	
3.2. Types of Projects	
3.2.1. Technical Projects	
3.2.2. Business-Oriented Projects	
3.2.3. Research Projects	
3.2.4. Entrepreneurial Projects	
<b>4. Managing Project Execution</b>	<b>7</b>
4.1. Group Formation and Size	
4.2. Group Member Roles and Contributions	
4.3. Project Selection and Proposal	
4.4. Project Supervision	
4.4.1. Report Submission	
4.5. Software Development	
4.6. Presentations and Demonstrations	
4.7. Course Timeline	

<b>5. Project Assessment and Grading</b>	<b>10</b>
5.1. Interim Reports	
5.2. Weekly Lectures, Meetings and Status Reports	
5.3. Mid-course Presentation	
5.4. Software Product	
5.5. Final Project Report	
5.6. Final Presentations and Posters	
<b>6. Resources</b>	<b>13</b>
<b>7. Useful Links</b>	<b>28</b>
<b>8. References</b>	<b>28</b>

## 1 Purpose of the Project Course

The courses in the BSc degrees at the DCIT focus on developing competencies in a specific area. The project course however utilizes all of the skills from these specific areas to solve one real world problem. Computing professionals often provide workplace solutions to stated problems based upon sound project management skills. Specifically, this course builds upon common concepts explored in previous system's development courses.

- **Aims and Goals**

The course aims to give students experience of working in small groups and participate in all the development phases (requirements analysis, design, construction, testing and documentation) of a nontrivial software system. It also gives valuable teamwork experience. The project assesses your ability to select a problem analyze and propose solutions then design, build and test a software system. It tests your ability:

- to make good decisions and recover from bad ones
- to utilize the collective skillset of the team to overcome various types of difficulty that may emerge
- to work within the constraints of limited resources; particularly time
- to document design decisions and communicate technical concepts

- **Course Delivery**

Groups of students will liaise with an academic supervisor. Presentations will be conducted on selected topics from project management and research methodologies. The course will be delivered using weekly consultation sessions with student groups (refer to course's timetable for more details). Assessments will take the form of attendance at meetings, written deliverables, presentations and course participation.

- **Learning Outcomes and Objectives**

While no two projects will be the same, all projects will test students in the same way such that upon the successful completion of this course, students will be able to:

1. Prepare a project plan for a technology-based project following standard system development methodology.
2. Make and defend sound software design and development decisions
3. Manage a project from start to finish meeting stated milestones and timelines

4. Write a project report outlining methodologies, group responsibilities, project milestones and lessons learned.
5. Demonstrate interpersonal skills, teamwork, and effective use of appropriate technology associated with system planning and development.
6. Demonstrate effective written and oral communication techniques through the assessment components (detailed in later sections).
7. Discuss and evaluate their contribution to a group project.
8. Review and provide feedback on other group projects.

## 2 Prerequisite Courses

A student enrolled in the project course should have successfully completed a course on Information Systems Development or Software Engineering.

## 3 Project Descriptions

- **Project Scope**

The scope of an undergraduate final year project should clearly define the problem or opportunity that is being addressed. The extent to which a software solution is fully developed depends on the type of project being undertaken.

- **Types of Projects**

Students may select a project idea that falls within one of the following four categories:

- Technical Computer Science or Information Technology Projects
- Business-Oriented Computer Science or Information Technology Projects
- Research Projects
- Entrepreneurial or Student-Proposed Projects

### 1.●.1. Technical Projects

A technical project focuses on producing clean, well-factored, error free code in a software solution that addresses a particular problem. This type of project should produce a good (best) software solution to a problem, but does not have to prove it is the best solution. A technical project may still apply to a particular domain and have practical uses. **A good breadth of algorithmic knowledge is imperative.** Students attempting this type of project

should have the necessary algorithmic programming skills in the area of application, and should have reasonably good knowledge of related areas as well.

### 1.●.2. **Business-Oriented Projects**

A business-oriented project focuses on producing either a customized software solution or an evaluated design in response to a common problem in a specific application domain. This type of project may involve identifying and integrating suitable software components, or configuring platform technologies or toolkits to fulfill a set of business requirements. A business-oriented project may produce a cohesive software solution featuring clean, well-factored, integrative code. **Integrative software engineering and critical thinking skills are essential.** Students attempting this type of project should have the necessary knowledge and skills using computer-aided software engineering tools, and should have reasonably good knowledge of the business domain as well.

### 1.●.3. **Research Projects**

A research project focuses on conducting work on a research topic under the supervision of an academic staff member. This type of project typically requires developing a testable hypothesis, involves background reading, and requires systematic evaluation of a particular strategy or solution for a specific problem or topic applicable to the research area of interest. A research project provides a valuable introduction for students interested in possibly moving into a postgraduate research degree and may result in collaborative publications with the project supervisor. **Strong scientific writing skills and analytical skills are essential.** Students attempting this type of project are expected to learn the appropriate research methodologies and techniques necessary to support their project, clearly present their work and its contribution in context of the current literature.

### 1.●.4. **Entrepreneurial Projects**

Students are encouraged to propose ideas of their own and choose a project that meets with the approval of the course coordinator. This type of project is best suited for students with a start-up idea for a company. An entrepreneurial project requires a business sense for marketing the idea, an ability to anticipate the future business need for the product/service, technical expertise to produce a prototype, and an appreciation of its social context of use. **Problem-solving, creativity, and lateral thinking skills are**

**paramount.** Students attempting this type of project are expected to defend the viability and novelty of their idea, produce a working prototype, assess the business risks, and present their business plan.

## **4 Managing Project Execution**

Project management deals with organizational matters that are needed for effective teamwork. An identified group leader can take the lead in the coordination of project activities, or the group can decide to share coordination responsibilities. Project management includes the following:

- Organizing group meetings and keeping track of deadlines
- Managing shared resources (such as documentation, software repository, etc.)
- Integrating individual contributions in a coherent manner, and resolving ambiguities and conflicting information including: collating different sections into a project report representing the work of entire group
- Ensuring that the project blog and reports are proofread, collated to ensure consistent layout, font styles, sections, and language style
- Integrating different parts of the program into one software system
- Running integration tests and ensuring that the whole system works as intended
- Fixing software bugs that result from incompatibilities between software modules contributed by different group members
- Other matters relating to the group's dynamics

### **4.1 Group Formation and Size**

Students are expected to choose their own group members from amongst the students registered for this course. A group size of 3 is recommended. However, exceptions may be made for large complicated projects, pending approval from the course supervisor. Group sizes exceeding 4 will not be permitted.

### **4.2 Group Member Roles and Contributions**

Project groups often function more effectively when group members have designated roles. Three core roles and responsibilities are:

- **Project Lead:** responsible for keeping the group on task, distributing the workload, meeting deadlines and ensuring smooth group communication and coordination

internally as well as accountability with the course coordinator and project requirements.

- Documenter Lead: responsible for recording group discussions and decisions, documenting various aspects of the project's progress, and ensuring well-formed reports and project documents are produced.
- Development Lead: responsible for overseeing the collaborative software aspects of the project, troubleshooting major technical problems related to code or integrative software, and ensuring code reviews, merges and testing are done well.

The entire project team should be engaged in discussions, documentation and development of the project. All members are expected to contribute towards the project. Groups will also have the option to rotate the roles among members for different stages of the project. This will allow members to gain experience through being responsible in different areas of project management.

### **4.3 Project Selection and Proposal**

A list of available project ideas will be made available through the course coordinator. Students are advised to start exploring their options early, and to select a suitable idea that is of interest by the designated date. Alternatively, students may propose an idea of their own. Proposed project ideas require approval by the course coordinator. **Only ideas that have been approved can proceed further.**

### **4.4 Project Supervision**

Projects will be supervised by the course coordinator. A Teaching assistant will be available for closer consultation, assistance and one-on-one advice with groups. A co-supervisor (an identified university staff member) may be assigned for selected projects; based on the nature of the project and the availability of resources.

### **4.5 Software Development**

Groups are encouraged to adopt agile software development methods as the practice is relatively simple to adopt and provides greater project visibility, predictability and flexibility. It also encourages development of high performance teams and early prototype.



A GitHub repository is required for each group project to facilitate project management and keeping track of software and design changes. The repository will also be used by the coordinator, research supervisor and teaching assistants to monitor, review and assess components of each project throughout the course. Note: students should create their own GitHub accounts if they do not have one.

#### 4.6 Presentations and Demonstrations

Interim presentations are intended to be formative assessments: demonstrations and brief status reporting will enable students to receive feedback on their work. The interim presentations will also be used to give and receive peer feedback. Peer reviews/critiques are intended to encourage lateral thinking and the development of review skills. Marks will be awarded to encourage participation in these activities.

#### 4.7 Course Timeline

Week	TOPIC
1	Introductions and 3 minute Elevator Pitch
2	Planning
3	Requirement Analysis, Milestone 1 (Documentation of previous processes)
4	Project Design
5	Project Implementation
6	Milestone 2 (Documentation of previous processes)
7	Project Implementation (Test Cases)
8	Project Implementation
9	Milestone 3 (Documentation of previous processes)
10	Project Implementation
12	Final Project Submission
13	Project Closing

## 5 Project Assessment and Grading

A detailed breakdown of the assessment components and their weightings follow below:

### **Project Report and Demonstration - 80%**

- Software
  - o Code Structure
  - o Working Software
- Documentation
  - Project management documents
    - o Status Reports
    - o Work breakdown and time line
  - Interim Project Reports
    - o Project Scope Document
    - o Project Timeline and Use cases
    - o Implementation Document
  - Final Project Report

Note: There **Project Report and Demonstration** submissions are categorized:

- (1) **Medium Fidelity Prototype** (Software 40% and Documentation 40%)
- (2) **High Fidelity Prototype/Final product** (Software 50% and Website and Documents 30%)

### **Presentation - 20%**

- Interim Presentation and course participation – 10%
- Final Presentation on Video – 10%

## 5.1 Interim Reports

Three interim reports are required during the semester:

- A project scope document (Milestone 1)
- A project timeline and use cases document (Milestone 2)
- An implementation document (Milestone 3)

## 5.2 Weekly Lectures, Meetings and Status Reports

*Weekly Lectures:* 1 hour per week. Relatively little of this time will be spent in didactic lecture. These sessions will reinforce/review project-based concepts and skills as needed. Class time will focus on addressing issues relevant to weekly meetings, and group activities, such as providing peer feedback.

*Meetings:* Groups are required to meet once per week with the course coordinator to provide updates on the progress of their project. Apart from offering technical advice the job of both the course coordinator and technical consultant is to check that you are working properly as a group and that you are/can get on target to meet deliverables. The status of your progress (planned vs actual) must be kept up to date on your website. Also, the link to the Weekly Status Report must be shared *prior* to the meeting. **Attendance of all group members at all meetings count in the assessment.**

## 5.3 Mid-course Presentation

Approximately mid-way through the semester, students are expected to give a high-level showcase of their project to the class. This is in preparation for the final submission, to share the current work being done with the groups, and to receive feedback. This presentation summarizes the interim prototypes and reports submitted up to that point, and expands on important points related to the following questions:

- What is the problem that your project addresses or the opportunity that your project takes advantage of? (Domain area)
- Who will be the main beneficiaries of your project? (Target audience, users)
- How significant is your project solution? (Novelty/usefulness of your project)
- What are the goals and objectives of your project (Deliverables, prototype)?
- What does your system prototype look like? (Architecture diagrams, Screenshots, Demo, Technologies used etc.)
- Where has your group reached so far? (Design, Implementation, or Testing stages)

#### **5.4 Software Product**

The software component of each project will be assessed as follows:

- Documentation (10%): Is there a Readme file? Are the modules explained clearly and sufficiently? Are there installation instructions or required software listed for complex systems?
- Robustness (10%): Does the code break for different test cases? Is the input data thoroughly validated?
- Modularity (10%): Is there sufficient abstraction and separation of input, processing and output components? Is there evidence of Design Patterns or MVC?
- Scalability and Efficiency (10%): Is it easy to extend the functionality of the software? Are efficient algorithms used? Are there any data or memory leaks?
- Meets Project Goals (10%): Are all of the required features accounted for in the final product? Is there sufficient complexity in the final solution?
- Working Solution (30%): Does the system work completely? Are there failures for critical components? Are there failures for minor components?

- **Student Code (20%):** How much of the code did the students write themselves?  
How many repurposed modules have been used? How much open-source packages have been reused?

## **5.5 Final Project Report**

The final project report aggregates the interim reporting with additional information about the project. A detailed description of the report format and content is included at the end of the manual.

## **5.6 Final Presentations and Video**

The live final presentation in week 13 elaborates on the mid-course presentation with accompanying video demonstrations of the working product. The video should be a maximum of 5 minutes long with narrated descriptions of the features. The project website should log all documentation intermediate prototypes, contain reference to the code and videos. All groups are required to submit a poster summarizing the work done.

## **6 Resources**

Templates for the following documents are included in the remainder of this manual:

1. Project Scope Document
2. Project Timeline, Work Breakdown and Use Cases Document
3. Implementation Document
4. Weekly Meeting Record Template (for academic supervisor)
5. Status Report Template
6. Mid-Course Presentation Guidelines
7. Final Project Document Template



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**

**Project Proposal Document**

TOC

1. Introduction
2. Positioning
  - 2.1 Problem statement
  - 2.2 Product position statement
- 3 Stakeholder Descriptions
  - 3.1 Stakeholder Summary
    - 3.1.1 User stakeholders
    - 3.1.2 Non-user stakeholders
  - 3.2 User Environment
- 4 Product overview (where applicable)
  - 4.1 Product Perspective
  - 4.2 Needs and Features
  - 4.3 Alternatives and Competition
- 5 Other Product Requirements (where applicable)
  - 5.1 Browser Compatibility
  - 5.2 Usability
  - 5.3 Responsiveness
  - 5.4 Visual design
  - 5.5 Long term Scalability



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Project Scope Guidelines**

**Project Scope Statement**

The project scope is a short document prepared primarily for the client. The scope statement clearly describes what the project will deliver and outlines generally at a high level all the work required for completing the project.

-----

Project Name:

Group Members:

Project Scope Statement

General Project Information

Problem/Opportunity Statement:

Project Objectives:

Project Description:

Business Benefits:

Project Deliverables:

Estimated Project Duration:



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Project Course**  
**Project Timeline, Work Breakdown and Use Cases Guidelines**

Project Name: <State the name of your project>

Project Members: <List members>

Project Objective(s): <Two line statement describing your project's high-level objective(s)>

**Work Breakdown Structure - Deliverables**

1. Identify the deliverables (in your scope statement) to be produced in your project. This highlights the work to be done.
2. Decompose each large deliverable into a hierarchy of smaller deliverables. This involves taking a deliverable and breaking it down into lower and lower levels of detail.
3. The lowest level of detail is called a 'work package' which consists of activities and tasks.

**Timeline Development - Schedule**

1. Identify the activities and tasks needed to produce each work package.
2. Identify resources for each task (e.g. time, knowledge, monetary costs)
3. Estimate how long it will take to complete each task. Consider constraints - resources, time, knowledge
4. Determine which tasks are dependent on other tasks and develop a critical path.
5. Develop a schedule of all activities and tasks - weekly and monthly. Work out when each task is scheduled to begin and end. Use a Gantt chart.

**Due on <insert date>**

**Submit an electronic copy of your document on myElearning.**





**The University of the West Indies**  
**Department of Computing and Information Technology**  
**COMP 3990/INFO 3490 Project Course**  
**Project Implementation Document Guidelines**

Your project group is required to submit a document outlining the project's implementation details.

Project Name: <State the name of your project>

Project Members: <List members>

**- Main Deliverables -**

- 1) Architectural Design: Component diagrams, high-level descriptions of the components in your system, and their purpose in relation to the project's objectives.
- 2) Class Diagram: Outlines the attributes, methods and interactions of the major classes/modules in your system.
- 3) Entity Relationship Model: Specifies the entities, datatypes, and relationships that are important for the project domain.
- 4) Sequence Diagram: Illustrates object interactions for the major use case scenarios for the project.
- 5) Description of Technology Used: Hardware devices, software products, programming languages.

Ensure that your diagrams follow proper modelling conventions. Include appropriate writeup to connect the sections. Overall, your document should have a smooth flow, logical transitions, and should be easy to follow.

**Due on <insert date> Submit an electronic copy of your document using myElearning.**



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Weekly Meeting Record**

<For Coordinator Use>

Project Name:		
Group Members:	1) 2) 3)	<input type="checkbox"/> Present <input type="checkbox"/> Present <input type="checkbox"/> Present
Date:		
Meeting #		<input type="checkbox"/> On Time
Documents Submitted	<input type="checkbox"/> Status Report Time	<input type="checkbox"/> On
Issues Group Working On		
Assessment of Progress	<input type="checkbox"/> Excellent <input type="checkbox"/> Good <input type="checkbox"/> Satisfactory <input type="checkbox"/> Fair <input type="checkbox"/> Poor	
Notes/ Concerns/ Comments		



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Weekly Status Report Template**

**Project:** <the name of your project>

**Date:** < dd/mm/yyyy>

**Iteration:** <the version/iteration of the project/prototype>

**Implementation Status**

Briefly describe the prototype or what can currently be demonstrated by the system; i.e. what is now available for feedback from the customer and testing within the development team? Attach or include a few sample screen shots to illustrate the progress.

**Highlights**

List any items of note. Breakthroughs, accomplishments, major decisions, or changes in the project plan. Are you on schedule, ahead of schedule or behind schedule?

**Risks or Issues List**

In the following table, list any risk or issue that is critical for the success of the project. This could be anything from "*we need to get test data*" to "*how do we ensure that the system is usable*" to "*performance is unacceptable*". This should be a complete historical list that is kept from the beginning of the project until the end.

*Status* should be one of *New, Ongoing, Closed*.

The resolution column should be filled in if the issue or risk has been taken care of.

A project may be expected to have around 1-3 active issues or risks that are being managed ( New or Ongoing ) at any given time. If you have more than three, then either you have a project in serious trouble or your criteria for what is "critical to success" is too loose.

Date Entered	Risk or Issue	Description	Resolution	Status


**Tasks in Progress or Completed in the Last Iteration:**

List the tasks that each member of the project worked on up to the present time.

Task Name	Description	Team Member Responsible	% Complete

**Upcoming Tasks for the Next Iteration:**

List the tasks that each project member is planning to work on in the upcoming iteration.

Task Name	Description	Team Member Responsible



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Mid-Course Presentation**

**Presentation Guidelines**

- Live and Video presentation required:
  - Introduction: Project name, Group members.
- **Video link must be submitted on website**
- Maximum 5 minutes.
- All group members must participate (speak) in the presentation in order to get marks.
- Business casual wear is appropriate

**Presentation Content**

Your presentation should be a high-level showcase of your project. Review your scope statement document for important points related to the following questions:

- What is the problem that your project addresses or the opportunity that your project takes advantage of? (Domain area)
- Who will be the main beneficiaries of your project? (Target audience, users)
- How significant is your project solution? (Novelty/usefulness of your project)
- What are the goals and objectives of your project (Deliverables, software )
- What does your system prototype look like? (Architecture diagrams, Screenshots, Technologies used etc.)
- Where has your group reached so far? (Design, Implementation, or Testing stages)

**Grading**

Your group presentation will be graded on:

- Organization, timeliness, and quality of slides/visual aids.
- Presentation content, mastery of domain, technical rigor of project
- Presentation and language skills.
- Overall impression and professionalism.



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Final Project Document Template**

**Total (100 marks)**

**1. Cover sheet**

- Name of Project
- Team Members
- Course Number, Course Name
- Department
- Lecturer
- Date Submitted

**2. Abstract (5 marks)**

A short abstract (300 words max) should open the paper. The purposes of an abstract are:

- To give a clear indication of the objective; scope, and key result of the project so that readers may determine whether the full text will be of interest to them.
- To provide keywords and phrases for indexing, abstracting and retrieval purposes.

The abstract should give synopsis of the problem, outline the approach taken, and present the key results or tasks accomplished. The abstract should not attempt to condense the whole subject matter into a few words for quick reading.

**3. Introduction**

- Problem Description (5 marks)  
Describe the problem or need that the team has addressed. Identify the purpose/objective of the project, the context and the general technical problem the team was solving. Describe tasks performed. Include information about any research analysis undertaken. Identify resources required, major risks, task schedule and major milestones.
- Positioning (5 marks)

- Stakeholder Descriptions (3 marks)
- Product Overview (5 marks)

#### 4. Requirements Specification (20 marks)

- Functional
- Non-functional (Quality attributes)
- User Stories:
- System diagram
- Use Case Diagram
- ERD
- Technical constraints

Use appropriate citations in the team's work. Citations document the sources that underpin concepts, positions, propositions and arguments. Citations reveal the quality of work that supports a discussion. Always cite:

- Statistics or specific numbers
- Statements made by specific individuals, companies or reports.
- Opinions made by others

Citations demonstrate that a position or argument is thoroughly researched. The report should give proper attribution to those whose thoughts, words and ideas are used.

#### 5. Design Specification (20 marks)

- System Architectural Design
  - Chosen System Architecture
  - Discussion of Alternative Designs
  - System Interface Description
- Detailed Description of Components/Subsystems
  - Component 1- n
- User Interface Design
  - Description of the User Interface
  - Screen Images
  - Objects and Actions

Specification and requirements for the project, include the detailed specification that serves as the basis for the project as an appendix.

Include expected customer requirements and desired features related to form, fit, function and interfaces. Consider aspects such as cost, safety, user friendliness, performance, convenience, misuses, capacity, legal issues, standards or codes, reliability, availability, maintainability, materials, productivity enhancement, entertainment, technology and design methods.

## **6. Implementation (Approach and Methodology) (20 marks)**

List the technologies/libraries, languages used in project.

Include Navigation map and screenshots of the system

## **7. Test (3 marks)**

- Test Plan
- Test Approach
- Features Tested
- Features not Tested

Describe what constitute project success and why? Discuss the product/service tests that will confirm the project succeeds in doing what it intended to do. Use the use cases or design verification matrix to determine test cases.

## **8. Business Aspects (2 marks)**

Discuss the novel aspects of this service or product. Address why a company or investors should invest money in this product or service.

- Briefly describe the market and economic outlook for the industry.
- Highlight the novel features of your product/service.
- How does your product/service fit into the competitive landscape? What's out there? How does this differ?
- Describe IP or Patent issues, if any?
- Who are the projected clients/customers?

## **9. Individual contributions (2 marks)**

Describe in detail what each person did play in the project effort.

## **10. Financial Considerations (3 marks)**

- Project budget
- Cost projections needed for either for profit/nonprofit options.
- Financial plan (optional)

## **11. Conclusions, Lessons learned and Recommendations (5 marks)**

- Describe state of completion of project.
- Summary of Feasibility
- Future Work
- Outline how the project may be extended



**12. References (2 marks)**

**13. Appendix**

**14. Submit links to shared folder with project files, code repository and Final Project Report on Website and mylearning.**



**The University of the West Indies**  
**Department of Computing and Information Technology**  
**Undergraduate Project Course**  
**Final Project Presentation Video**

**Presentation Guidelines**

- Video demonstrations required:
  - Introduction: Project name, Group members.
- **Links to video must be submitted on myElearning by deadline date.**
  - Updates are allowed until week 12
- Maximum 10 minutes for your presentation.
- All group members must participate (speak) in order to get marks.
- Business casual wear is appropriate

**Presentation Content**

Your presentation should showcase your project from technical as well as a business perspective. Review your scope statement document for important points related to the following questions:

- What is the problem that your project addresses or the opportunity that your project takes advantage of? (Domain area)
- Who will be the main beneficiaries of your project? (Target audience, users)
- How significant is your project solution? (Novelty/usefulness of your project)
- What are the goals and objectives of your project (Deliverables, software )
- What is your system's design? (Architecture Diagrams, Technologies used etc)
- How does your system work? (Sequence diagrams, demo videos, live demos)
- How did you test your system? (Experiments, test data, user evaluation)
- What are your future plans for the system? How can it be extended?

**Grading**

Your group presentation will be graded on:

- Organisation, timeliness, and quality of slides/visual aids.

- Presentation content, mastery of domain, technical rigour of project, business appeal.
- Presentation and language skills.
- Overall impression and presence.

## 7 Useful Links

### Team Formation & Building

[Cyriac Roeding: Keys to Creating Awesome Teams](#)

[Choosing your Project Team members](#)

<https://agilewarrior.wordpress.com/2009/11/27/the-drucker-exercise/>

<https://www.productplan.com/5-ways-build-relationships-product-development-teams/>

<http://tobeagile.com/2012/12/11/seven-strategies-for-team-swarming/>

### Project Management

<http://www.littlethingsmatter.com/blog/2010/08/23/you-cant-improve-what-you-dont-measure/>

Git <https://github.com/>

**Elevator Pitch** - <https://fi.co/madlibs>

### The Agile System Development Life Cycle (SDLC)

<http://www.ambyssoft.com/essays/agileLifecycle.html>

<http://agilemodeling.com/essays/initialArchitectureModeling.htm>

<https://www.stickyminds.com/article/role-testers-agile-environment>

<https://wiki.fluidproject.org/display/fluid/User+Testing+Scenarios+and+Tasks>

### User Interface Design Tips, Techniques, and Principles

<http://www.ambyssoft.com/essays/userInterfaceDesign.html>

**UML Diagrams** <http://agilemodeling.com/essays/umlDiagrams.htm>

**Building Real Software** <http://swreflections.blogspot.com/2013/01/hardening-sprints-what-are-they-do-you.html>

**Read me files** <https://github.com/matiassingens/awesome-readme>

## 8 References

<https://www.skorks.com/2010/03/the-difference-between-a-developer-a-programmer-and-a-computer-scientist/>

<https://www.cs.cornell.edu/masters/entrepreneurism>

Albayrak A.(2017) [http://thesai.org/Downloads/Volume8No11/Paper\\_11-](http://thesai.org/Downloads/Volume8No11/Paper_11-)

[Restructuring\\_of\\_System\\_Analysis\\_and\\_Design\\_Course.pdf](#) (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 11, 2017