Vanier College

Faculty of Science and Technology

System Development

420-436-VA

Team Projects

Distance Learning Version

J. Lebensold

Table of Contents

Overview

This course explores the process of developing real-world information systems. The team projects that are central to the course allow you to demonstrate that you have synthesized the materials covered in the courses leading up to this course and to have an opportunity to see how a real commercial system is developed. You will be introduced to methods and techniques currently being used to determine what a client wants and needs. You will also have the opportunity to analyze these requirements and design and build an application that includes a database for an actual organization. As well, you will experience what it is to be part of a development team. The general objective of the course is to simulate the process of developing computerized information systems that you are likely to encounter in any analyst or programmer work environment. The specific objectives are to learn about project management, information and requirements gathering, data modeling, information-level design, user interface design and prototyping. You will also learn how to both lead, and be a member of, a project team. The course consists of two projects: The development of a prototype database information system (the **development project**), and documentation about the process of building that system (the **documentation project**).

The course will be conducted using various remote communications environments and technologies that are currently being used to develop commercial information systems.

Background

This course draws heavily on material that has been covered in previous courses in the Computer Science program. You will find it useful to review the course material as well as the textbooks, assignments and notes from these courses. Furthermore, you will find it useful to refer to various textbooks and web sites covering the following topics:

- Systems analysis,
- Project planning,
- Version control,
- Database management (including determining primary keys, using SQL, and creating database queries),
- Entity-relationship modeling,
- Use cases,
- User stories,
- Model-View-Controller architecture,
- The unified modeling language (UML),
- Prototype development,
- Agile development,
- Testing,
- Good user interface design.

It is also expected that you are capable of writing a coherent technical report in proper English, formatting it correctly, and using footnotes and references as appropriate[1].

The course will expose you to tools and technologies that are currently being used commercially, many of which will be new to you. You are expected to learn how to use them on your own using on-line tutorials, web sites and the provided reference material. This is in accord with what you will find on the job: New tools, technologies and techniques are constantly being developed, and as someone working in this field you are expected to keep yourself constantly up to date – often on your own time.

Teams

Each student in the class will be a member of a development team for the duration of the course, experiencing the thrill of system development (it's like being on a roller coaster, exhilarating and terrifying at the same time). Team members will collectively work on the projects, and each student in the team will receive a grade based on their contribution to the projects and on the overall quality of the projects. Work should be divided equally among team members in whatever manner the team determines to be fair. Ideally, this should be decided by consensus, and not dictated by the team leader. Try to build a "cross-functional team", where you identify each other's strengths, and areas where you would like to develop skills[2]. This requires a bit of reflection on the part

[1] This document is a good example of proper formatting

[2] All team members fill out a "Skills Inventory" which will be shared amongst team members.

of the team members in order to identify everyone's strengths. You can then work cross functionally in order to build a shared understanding of the system's design and learn how to collaborate rather than to work in "silos"[3]. In order to provide a system that reflects the needs of different users and is technically sound, different members of the team will need to focus on specific aspects while maintaining an understanding of the whole.

Each team is composed of four to six members. In keeping with the situation in the "real world", team membership will be determined by the instructor (the "boss"). If one member drops out of a team, the remaining members are TOTALLY responsible for the completion of ALL team assignments.

Much of your work for this course will be done during team meetings, which you may hold either during lab periods or during your homework time. There will also be time available during some lecture periods in which you will be able to hold team meetings. You should make a distinction between team working sessions and team organizational meetings. The former means several team members work together to solve various problems, either for the documentation project or the implementation project. The latter are structured meetings where all team members discuss what needs to be done to enable both projects to move forward. Your individual success depends to a large extent upon how well your team functions. You are responsible for organizing your team to ensure equal workloads and smooth team operation.

It will be up to the team to determine the platform you will be using to conduct on-line team meetings. Microsoft Teams (available as part of Office 365 from the College) will be used for lectures and labs, and the instructor will be available at those times using Teams for consultations. Team only meetings may be conducted using whatever platform you choose or may be held in person if health conditions permit. Due to health concerns, the instructor will only be available using Teams and is not available for in-person meetings. Whatever platform you use, bear in mind that you will also need a way of communicating with your potential client as well.

Journals

There are **two** different journals to be maintained: A team logbook, and a personal journal. In general, these may be either electronic or on paper; however, since they are to be submitted periodically to the instructor for grading, they should be in a format that can be sent to the instructor as a PDF.

The **team** logbook records all decisions and activities taken by the team. It records what happened and who participated in team organizational meetings. Do not use it to record the working meetings. It is also a record of who was responsible for what, who did what, when things are due, and when certain events took place. For example, you would record all the client meetings, who was present at them, what the client said, and what was supposed to be done as a result of the meeting. The team logbooks must be properly formatted and submitted at times specified in the course schedule. Logbooks are graded on the extent to which they provide the required information. An example of a team log is provided in the supplementary documents for this course on Léa.

The other journal is a **personal** one. You will find the personal journal indispensable when it is time for you to do the peer evaluation of your colleagues for each deliverable and when you write your Learning Integration Assessment Report. Furthermore, if you develop the habit of maintaining a daily journal, you will also find it useful when you start working in industry, in particular when you are subject to performance evaluations.

In the personal journal you should record your own activities to refresh your memory and to document both how you and your team members are performing. Keep track of project and team events as they occur during the semester, as well as the amount of time you spend on the different parts of each deliverable in the project. Also, record the amount of the time you spend viewing lectures as distinct from time spent working on the deliverables, assignments and preparing for quizzes. You should be recording how much time you spend on each specific task. Also, record where you work on this course: at home, at the coffee shop, on the bus, etc. If

[3] An information silo is a management system incapable of reciprocal operation with other, related information systems. (Information silo)

you are communicating with others, note down the platform you are using (in person, Skype, Zoom, Teams, etc.)

It is expected that you will be working on the projects in this course on an almost daily basis, and so **your journal entries should be made on a daily basis**. Keeping track of the amount of time you spend on different activities is an important discipline to learn. Also include the total amount of time you spend on the course per day.

For example, do not write:

> Monday, worked on deliverable, lectures and did a quiz.

Instead, write

> Today I decided to forge ahead on the project even though my team members seem to be taking it easy.
>
> Tuesday, Mar. 29, 09:20 to 11:10 (1 h 50 min – at home): Designed order entry screen
> Tuesday, Mar. 29, 11:15 to 12:30 (1 h 15 min – at home): Reviewed lecture for quiz
> Tuesday, Mar. 29, 12:45 to 13:15 (30 min – at Pizza Hut using Zoom): Lunch with team member Johan; discussed order entry screen layout.
> Tuesday, Mar. 29, 13:30 to 13:45 (15 min – at home) Uploaded order entry screen to Github
> Tuesday, Mar. 29, 22:30 to 22:45 (15 min – at home using MS Teams) Took quiz
> **Tuesday, Mar. 29, Total time: 4 h 05 min**

Since this is a personal journal, do not include any details about the decisions taken during the team meetings. These will be recorded in the team logbook. You should, however, record how well you think the other members of your team are performing, and any other observations about the functioning of the team. These notes will be useful when doing peer evaluations for each deliverable.

This journal is a personal one, and the instructor will be the only one reading it. These journals will be examined periodically throughout the semester, so they must be kept up to date. They may be kept electronically or on paper; some students keep a diary, recording all events in their lives relating to their studies. However they are kept, a PDF of the journal must be available for the instructor to grade, when the instructor requests it. The grade will be determined based on the kind of content that is recorded and not the actual content; that is, the extent to which the journal follows the above guidelines.

Team functioning

One of the key goals of this course is learning how to both manage a team, and function as a member of a team. You will be provided with guidelines about how to do this, but it is your responsibility, both individually and collectively, to see that the team functions smoothly.

Both projects (documentation and development) are self-managed and self-directed by the team. The instructor's role is to provide advice and counsel, and to evaluate your work, but not to get involved with day-to-day team management, except in rare cases. It is your responsibility to stay on schedule, decide who will do what, and when, and to decide how to prepare the deliverables you create. In general, you must decide how to organize your team to ensure that the various parts of the project are completed on time. Furthermore, each team member is responsible for making an equal contribution to any group work. This requires personal maturity, professionalism, and commitment. Specifically, it is your responsibility to deal with personal matters within your team. There may be considerable diversity in experience, age, technical expertise, and other factors among the team members. Use this diversity to your advantage! Only when repeated attempts on your part to resolve disagreements or conflicts produce no outcome should you involve the instructor. The instructor will remain available for consultation about any aspect of the course.

Since your individual grade for each deliverable is moderated by the peer evaluations *(Figure 1)* done by your peers, if you receive a poor evaluation (resulting in a lower mark than that of your peers) it is important for you to find out how you can improve your performance for successive deliverables. This can be done by having a

frank and open consultation with your team members. In industry, it is common practice for managers' evaluations to also take into account evaluations done by your peers, whether formally or informally.

13. Using the scale below, evaluate this teammate, submit the form and then submit a new form for other teammates.

---

5 Excellent: Consistently went above and beyond; tutored teammates, carried more than his or her fair share of the load, present for every meeting; contributed to the highest degree; worked very hard
4 Satisfactory: Usually did what he or she was supposed to do, acceptably well prepared and cooperative
3 Ordinary: Often did what he or she was supposed to do, minimally well prepared and cooperative
2 Marginal: Sometimes failed to show up or complete tasks, rarely prepared
1 Poor: Often failed to show up or complete tasks, rarely prepared
0 No Show: No participation at all, was not present at all; did not contribute to effort at all; showed no effort at all *

	0	1	2	3	4	5
TEAM MEETINGS: Participation / attendance in team meetings	○	○	○	○	○	○
TEAM MEETINGS: Contribution to discussions	○	○	○	○	○	○
INTERACTIONS: Communicates clearly, with civility	○	○	○	○	○	○
INTERACTIONS: Listens effectively	○	○	○	○	○	○
INTERACTIONS: Accepts criticism gracefully	○	○	○	○	○	○
INTERACTIONS: Helps others when needed	○	○	○	○	○	○
TASKS: Completes fully	○	○	○	○	○	○
TASKS: Completes on time, or makes alternate arrangements	○	○	○	○	○	○
TASKS: Quality of writing	○	○	○	○	○	○
TASKS: Contributes to team effort	○	○	○	○	○	○

Figure 1: Peer evaluation form

A different member of the team will be responsible for each deliverable. This person becomes the team leader for that part of the project. When the teams are formed, the instructor will identify the team leader for each of the deliverables. Under exceptional circumstances, this assignment may be changed upon consultation with the instructor. The team leader is not a demi-god, but rather is the person who coordinates activities and through whom the instructor can contact the team. The team leader will also make a short (no more than five minutes) presentation about the work that was done during their tenure, following the submission of each deliverable, as per the schedule. *The presentation will be made using MS Teams*, normally in the lab or class period following the submission of the deliverable. The team leader is also responsible for ensuring that all submitted documents are properly signed by ALL team members (*electronically, if possible*), and that a PDF copy of each deliverable is sent by email to the instructor. **Team members who do not sign a deliverable, or fail to submit completed peer evaluations, forfeit their right to any marks for that deliverable, and will receive 0 for that deliverable**. All team members should have an email address by which all other team members can communicate. Deliverables submitted late will be penalized 10% per day.

System development and database design are iterative and recursive processes, so do not be deceived or frustrated by your experiences. It is natural to refine descriptions and designs throughout the project, and it is normal for things to go wrong. You should adopt an attitude of learning, which involves acting, reflecting on the actions taken, consulting on the results, and then moving forward based on the lessons you have learned.

Projects

As stated above, you will be working on two projects simultaneously: A development project, and a documentation project.

If at all possible, the development project should originate in a real-life company or organization. Suggested companies are the following: Bicycle stores, clothing stores, music stores, small private schools, dépanneurs, restaurants, small grocery stores, gyms, small private businesses, daycares, seniors' residences and so on.

What is important is that the owner or manager is willing to spend some time with you to describe how their business works and discuss with you business problems they may be having. Their feedback is critical. The more time you have with them the better, as long as you can capture requirements and break them down into user stories. You will be taught how to do this during your lectures. Your clients typically won't talk to you unless you have something to show them, but by making sure that the feedback loops are short, you can get more iterations in and end up with a higher quality product that is more suited to their needs. Value their time above everything else and be focused, have an agenda and prepare things ahead of time.

Finding a client

From the first day of class, you will be looking for a client.

In the past, students have found clients by talking and emailing members of their families, close friends and acquaintances, or by going door-to-door to businesses located either near the College or their homes. Since you will be developing software at a distance, the physical location of your client does not matter. A couple of teams in the past developed projects with international clients. While distance does introduce other interesting challenges, such as working in different time zones, this is consistent with contemporary software development.

When you are contacting a potential client, you should explain the setting; that is, that you are CEGEP students doing a class project. Your goal is to develop a prototype DB application that will help improve the client's business. You should emphasise that you will be developing this application WITH them, and not just FOR them.

Here is a sample email you could modify and use:

> *Dear [Uncle | Aunt] Vivian,*
>
> *As I think you may know, I am presently furthering my studies of Computer Science at Vanier CEGEP. This semester, one of the courses I am taking is System Development. As part of the course, I am part of a small team of students developing software to help solve a business problem. Ideally, the problem should come from a real company. I thought about **your** business and problems you might be having.*
>
> *{I remember you saying that you were having difficulty keeping track of [your customers | your orders | your sales | your suppliers | etc.]. This is exactly the kind of problem we may be able to help resolve.}*
>
> *OR*
>
> *{Businesses face challenges as they grow and develop. The recent emergency measures have created both new opportunities and new difficulties for many businesses. I think that this may be the case for your business. If so, then our team may be able to help.}*
>
> *Since the resulting software is part of a course, there would be no cost to you.*
>
> *Because of the present health emergency, it may be difficult to meet in person. I would like to speak to you to explain in more detail how this collaboration would work, and to explore the kinds of problems and potential solutions we could address. When would be a good time to [telephone | FaceTime | Zoom] you in the next day or two?*
>
> *Sincerely yours,*
>
> *[Your name, your email, phone, etc]*

In trying to find a suitable client, focus on their **business** needs. Almost any business can benefit from a database application, but they may not realize it. You should be able to explain to them how using a database will improve their day-to-day functioning, and possibly their bottom line. Use your knowledge of database systems, but in most cases, it is probably better to **avoid talking about technical issues**. Reassure them about the ease of use of the system you will develop, and of its suitability to their business. You should also explain to them that you will be delivering a prototype of a database system that will answer some of their needs, not a full-blown application that will solve all their information processing needs.

You can start by talking about the business processes that are causing the most waste in their organization, leaving the actual database design until later. A process could be “entering a new bicycle into inventory” or “handling a customer refund” or “creating a new customer” or “manage my inventory” or “keep track of my customers”. A database will be critical to the success of the project, but it need not be the focus of the business problem. This approach will also help the business identify routine processes and organize their systems accordingly.

In general, it is preferable that you do **not** work for the company or division that becomes your team’s client. All the members of the team will need to be able to interact with the client at different times, so the company needs to be open to dealing with different members of the team.

Ideally, the organization should have more than one employee, so that your team can get a more realistic experience with working with several different people.

If you are unable to find a real organization, you will have to do a project based on a fictional organization, assigned by the instructor. In this case, the instructor will act as your client representative for the duration of the project. In the past, students have found this option much more difficult.

Software, Hardware, and Technical Assistance

It is recommended that you complete the development project using software that is available to you through the College. You are encouraged to download and install the complete MS Office Suite. It is, however, your

choice as to which software you will use for any given part of the project. In particular, it is up to you to decide on a DBMS: some version of SQL, Oracle or something else, and a programming language or environment. Whatever you choose, be prepared to explain your choice. Remember that you will have to develop a good user interface as part of the project, so pick an environment that is flexible and relatively easy to use.

One of the key success factors in a project of this nature is the ability for several people to work on different aspects of the project at the same time. In order to facilitate this, commercial enterprises use version control systems (VCS). Use Git or a variation of it. You may also use any of a number of applications available on the Web. Before deciding which tools you will use, you should do adequate research to determine whether those tools can answer your needs, and whether you can master them sufficiently in time to produce the required output. You should explore using Ruby on Rails or the Zend Framework, as alternative environments in which to develop your application. Students have found it challenging to develop a non-trivial, friendly user interface using the .NET framework. You may decide to use MySQL Workbench or Oracle SQL Developer. You should avail yourself of the many free on-line tutorials on these subjects. There is also some material posted on the course Léa site.

The following are recommendations, and have produced good results in the past:

- Ruby on Rails with MySQL for implementation, using CSS and HTML for the user interface
- Git and GitHub for version control, and document and code sharing
- MS Project or ProjectLibre for project management
- MS Visio or an open source application such as Lucidchart for diagrams.
- MS Word for reports
- MS Excel for analysis
- Trello or Miro for User Stories and story mapping

Project Deliverables

For the documentation project, each team will submit the following deliverables, on the dates shown on the course schedule:

Deliverable		Value
1.	Project plan	10%
2.	Client and business domain summaries, questionnaire	10%
3.	Analysis - Use cases, UML Diagrams	15%
4.	Requirements - User stories	15%
5.	Prototype and client comments	15%
6.	DB design	15%
7.	Implementation and client comments	20%
Total		100%

The project grade will be based upon an evaluation of the project deliverables. The grade received will be for both content and form. In general, it is a good idea to enhance the professional appearance of your product. In particular, everything you submit must be free of grammar and spelling errors. Please note that grammar and spelling correctors do not catch all errors. You may wish to use this document as a model for the design of your deliverables.

The team is to turn in one copy of each written deliverable for the project, **emailed** as a PDF (Word, or other .doc formats, will NOT be accepted; submissions sent through Omnivox will NOT be accepted). The electronic

copy will be graded and returned to the email address from which it was submitted. The dates and times for submission of each course deliverable appear on the course schedule.

Each document handed in will have a cover page, a table of contents and an executive overview.

- The cover page contains the name of the team, the name of the deliverable, the date of submission, the names of the team members who contributed to the deliverable, with their signatures, and the name of the team leader for the deliverable. Use this format:

  I, (firstname lastname), student ID# (ID number), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).
- An **executive overview** (maximum 1 page) which summarizes the major findings of the project to-date and highlights (in words) the elements of the report being submitted. An executive overview is more than a table of contents in narrative form. It actually summarizes the contents of the document and contains all the essential information a business executive who does not necessarily have time to read the whole document needs to know to understand the crucial elements of your project at that point. Typically, it is written after the rest of the report is completed.
- Here is an example of an executive overview from a previous team project (Deliverable 4):

  In this deliverable, we go over our plans for our future information system. To do so, we employed an Agile software development tool: user stories. We created thirty user stories based on our client's current system and developed it further to reach our future information system goals. A detailed narrative description of our client and our future information system is included.

  We list all our planned new features, i.e., user stories, in order to achieve a concrete procedure when we start our implementation. User stories need confirmation that they can be implemented correctly and that they follow the given conditions that each should satisfy. For that purpose, we made sure to create a minimum of two tests for each user story which follow the GIVEN-WHEN-THEN format for ample clarity.

  In the last section of this deliverable, in order to showcase the project backlog, we have also enclosed a user story map. This allows a visual representation of the user stories which aids in the understanding of the functionality of our system.

If for any reason (other than procrastination or mismanagement by your team) you need to renegotiate the course deliverables or their timing, this must be done in advance of the due date (***in writing by email***) with mutual agreement by the instructor and all team members. Otherwise, late course deliverables will have the grade reduced by 10% for each working day (24 hours) late.

As your prototype develops, you will continue to interact with your client. Throughout this process, your client will make comments and suggestions about improving the user interface and the functionality of the prototype. Keep a record of these comments and suggestions **in the team logbook** and submit them when requested as part of the various deliverables.

Before starting work on these deliverables, read through this entire document and make sure you and your team understand what is required in each deliverable. If anything is unclear, ask the instructor.

Remember that a **Peer Evaluation** from each member of the team must accompany each deliverable and the implementation. **If a team member does not submit a peer evaluation at the same time as the deliverable, that team member will receive a mark of 0 for that deliverable**.

1st Deliverable – Project plan (65 marks)

One of the first things that a project team needs to do is to organize itself. This means determining who is going to do what, when they are going to do it, and who is going to take responsibility for the different deliverables, and parts of deliverables. In order to do this, there must be a definition of the sequence of events that will result in the deliverables being delivered on time. This is called a project plan. In this deliverable you will provide a project plan for the documentation project, which is concerned with the *process* of building the prototype information system.

At this early stage in the development of the project the plan will be very general; however, by looking at the activities, the deliverables, and the resources at your disposal, you should be able to put together a coherent plan *for the rest of the semester* identifying when the deliverables will be submitted, who will work on the various parts, and how long they will take to do each part.

In preparing your project plan, be sure to take into account the various deliverables, any planned absences, as well as the capabilities of the team members. Note that some deliverables require the completion of others, while some do not, and can be worked on in parallel.

Because of the nature of the documentation project, it is possible to plan its execution without actually having a client. In order to complete future deliverables, it will be necessary to have secured a client.

This deliverable should contain the following information:

- (10 marks) **Front matter** (this will appear in **each** deliverable)
    - Cover Page including: (6 marks)
        - (1 mark) project title,
        - (1 mark) date,
        - (1 mark) team name,
        - (1 mark) names of all team members,
        - (1 mark) name of client/sponsor organization, (if available)
        - (1 mark) client contact name(s), (if available)
    - (2 marks) A statement about using previous work in the deliverable. It is possible that your project will use code and ideas that you developed in another course, or at your place of work. Identify any previous (or concurrent) course project work on which your current project will build. If none, say so.
    - Table of Contents. (2 marks)
- (5 marks) Executive Overview
    - An executive overview (maximum 1 page) summarizes the major findings of the project to-date and highlights (in words) the elements of the report being submitted. It is more than a table of contents in narrative form. It actually summarizes the contents of the document and contains all the essential information a business executive who does not necessarily have time to read the whole document needs to know to understand the crucial elements of your project at that point. Typically, it is written after the rest of the report is completed.
- (10 marks) **If you have identified a client/sponsor**, then:
    - Brief description of the client/sponsor and potential user(s) to be served by the proposed system (3 marks).
    - Include the computer skills and literacy for the client/sponsor and potential user(s) (3 marks).
    - A description/statement of the **business problem** to be solved by the implementation project. (4 marks)

    OR
- (10 marks) **If you have NOT identified a client/sponsor**, then:
    - Brief description of the different potential clients you have contacted, and why they were unsuitable, or did not want your assistance. (1 mark for each one contacted, up to ten).

- (22 marks) Brief description of how your team is to be organized:
    - (2 marks) Regular team meetings (when and where, include a sample agenda)
    - (1 mark) Online repositories (which ones?)
    - (2 marks) Communications strategy (how team members are going to communicate with each other, what policies you have established).
    - (2 marks) When and how will you meet synchronously?
    - (13 marks) Areas of responsibility (**These must change for each deliverable**)
        - The team leader for each deliverable will be determined by the instructor
        - (6 marks) Client contact (which team member will be the primary client contact for each deliverable, after the first one)
        - (7 marks) Reports (who will make sure that the reports are prepared properly, and on time, for each deliverable)
    - (2 marks) Contact information
        - (1 mark) Email addresses for each team member
        - (1 mark) Cell or other telephone number for each member
- (11 marks) Project Plan, in the form of a Gantt chart that shows the steps you intend to follow for the remainder of the term, to complete the documentation project, in a timeline designed to meet the due dates for various deliverables. Enter as many detailed activities as you know. This chart and any accompanying description should clearly show for each project step the following items:
    - (2 marks) the task name and explanation,
    - (2 marks) estimated work time,
    - (2 marks) start and completion dates,
    - (2 marks) sequence and parallel nature of project steps, and
    - (2 marks) resource assignments for each step.
    - (1 mark) Include a PDF of the project plan as part of the deliverable.
- (3 marks) Make sure the entire report is correctly spelled and grammatically correct.
- (3 marks) Make sure the entire report is well formatted (appropriate headers and footers, suitable headings and sub-headings, consistent page numbers, etc.)
- (1 mark) Submit one PDF for the entire deliverable.

2nd Deliverable – Client and business domain summaries, questionnaire (53 marks)

Once you have identified your client, it is important to find out as much as possible not only about the specific business you are dealing with, but also about both the business domain and the business environment in which your client operates. This research activity[4] should start before you even meet the client for the first knowledge acquisition session. Is this a B2B or B2C? Do they provide a service, a product, or something else? How does the domain function? For example, is it increasing in sales/volume or not? How much, and what is the nature of, competition? Do businesses typically have an Internet presence? How widespread is the use of information technology?

If, for example, your client is a bicycle store, you should find out how the bicycle sales and rental business works: where do the bicycles, accessories, spare parts, and so on come from? How much do they cost? How often do new ones appear? Where are competitors located? What about on-line bicycle sales and rental businesses? Are there specialty items? In other words, you should become as knowledgeable as possible about the consumer bicycle industry. This is the business **domain**.

Similarly, you should try to find out as much as you can about the specific business with which you are dealing. Study their advertising, or their signage. If they have a web site, examine it. You might even consider taking photos of the outside of their business (don't do it inside – that is private property) and studying it. The objective here is to find out as much about the particular company before you actually talk to the client. For example: Where are they located? Who are their actual competitors? What sort of information is provided on their web site, or in printed advertising? This is the business **environment**.

Once you have gathered information about the domain and environment of your client's business you can start formulating a questionnaire. It is important to remember that this questionnaire will not actually be given to the client – it is intended to serve as a guide for you, for when you interview the client.

Your initial goal is to try to determine the business problem. What are the pain points in the business? Or, what do they see as the major challenge? Or, what keeps them awake at night?

Here is a statement of a business problem for a small restaurant (from previous student work, with actual names removed):

> *Starting a business is never easy and with being in such a competitive field every food shop has their flaws and ways of doing things differently. However, even though [client restaurant] has only been open for roughly 3-5 years, they have managed to do really well, especially considering they are located near so many competitive companies like [big chain restaurant] and [another big chain restaurant]. [Client restaurant] has developed successful strategies to adapt in their own ways and grow their own supportive clientele that really enjoys their food. Despite that, [client restaurant] has its own little difficulties, the main one being the way they keep track of their inventory. Currently, the way the owners keep track of their inventory is by estimating how much is needed rather than actually writing it down or making a spreadsheet using Excel. This results in running out of a certain product or throwing away unused items that are about to expire. This lack of proper inventory tracking can severely hamper the business. On the patron-facing side, it can lead to unpredictable changes in the menu, where favourite items can suddenly become unavailable leading to poor customer satisfaction and unfavourable reviews. On the book-keeping front, it can also prove highly disadvantageous, as it may lead to overspending, make it difficult to create a monthly or quarterly budget or, worse still, render it difficult to judge profit margins.*

In addition to questions about the general functioning of the business you could also ask questions like the following:

- How knowledgeable are you and your employees about IT?

[4] Make sure that you document the sources of your information, so that you can provide proper references in the text and a properly Works Cited section in your deliverable.

- What is their attitude toward computers?
- What software are you currently using (an Excel spreadsheet for bookkeeping counts as software)?
- What is the turnover rate for employees?
- How long does it take for new employees to become effective?
- How is the business organized?
- What sorts of roles exist within the business (people typically perform multiple roles)?

This deliverable should contain the following information:

- (3 marks) Front matter (see 1st deliverable)
- (5 marks) Executive Overview (see above for a description).
- (5 marks) Description of the business domain – this is a description of the industry, based on your research
- (5 marks) Description of the business environment – this is a description of the world in which your client operates, based on your research and observations
- (3 marks) Description of the client – especially computer skills and attitudes
- (5 marks) Business problem – what is the business problem that the new system will resolve?
- (10 marks) Open questions (things you don't know but wish you did). This should be a narrative; that is, it should read like a story. For the example above (the small restaurant), part of the narrative could read as follows:

  On the inventory-keeping front, we know it is being "roughly estimated" and that that method is defective and leads to poor performance. However, we do not currently know how that estimate is made or how frequently it is being conducted. We also do not have a good understanding of what exactly needs to be inventoried. We assume that meat, buns and toppings are highlights, but we recognise that there is much more to running a successful restaurant than the raw ingredients that make up the meals. There is cleaning to be done, for example, and takeaway boxes to be provided for that type of patrons. There are also perishable and non-perishable items. It therefore becomes crucial to determine what broad categories of items are needed as well as a rough idea of their respective turnover rate in terms of days, weeks or months.
- (10 marks) The questionnaire. This is a formalized version of the narrative. It will be an aid to finding the answers to the open questions. Continuing with the example, the questionnaire includes:

  How do you operate right now (as far as your inventory)?
  What different types of inventory do you have?
  How or where do you buy your supplies (e.g., distributors/stores)?
  How frequently do you update your inventory?
  Do you buy your supplies jointly with [another restaurant] location?
- (2 marks) Proper use of references
- (2 marks) Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting.

3rd Deliverable – Use cases and UML Diagrams (78 marks)

Before working on this deliverable, you must meet the client. It may not be possible to do this in person. If that is the case, then you will have to organize virtual meetings with your client. These can be conducted using MS Teams, Zoom, Skype, FaceTime, or even by telephone, depending on how comfortable your client is with these environments. In your first meeting, you can then start an analysis, using the questionnaire, as well as all the other tools and techniques mentioned in class. As part of this process, you will be able to identify the client's objectives.

After meeting the client, perhaps more than once, you will be able to document the results of your initial exploration and description of the existing business. Using use cases and UML diagrams build a model that describes the information system that **currently** exists in the business. Whether the owner keeps track of his stock on little pieces of paper, or in an Excel spreadsheet does not matter. What is important is to build a model of both the dynamic and static aspects of the information in the business. In addition to a system diagram and at least three detailed use cases, you should have at least one activity diagram and a sequence diagram for two use cases, as well as a class diagram for the entire system. The class diagram will be used later when you design the database.

This deliverable is a good test of your team's ability to work together, since the various diagrams required need to present a coherent description.

This deliverable should contain the following information:

- (3 marks) Front matter (see 1st deliverable)
- (5 marks) Executive Overview (see above for a description).
- (3 marks) Summary description of the client
- (5 marks) Business problem – what is the business problem that the new system will resolve. It may have changed now that you have a better understanding of how the present system works.
- (10 marks) Narrative description of the present information system – Do the diagrams in the appendices first, then write a text narrative that summarizes the diagrams. The narrative should read like a story describing what the various users do when they interact with the system. Here is an example of part of a narrative (from previous student work):

  The inventory is checked by the owner or by an employee who is qualified to do so, who then makes a list of the items that need to be ordered. The owner who obtains the list decides when to begin the order placement process.

  The manager or the member responsible for the inventory must check the inventory and notes items that are running low. Next, the supplier is called and is given the list of required items. The manager is then provided with a transaction number for the order. Upon the arrival of the new stock, the inventory, that is kept on paper must be updated.

  The time and date that items are prepared and displayed must be written on paper along with their expiry dates. This paper is hung on a wall and is then checked periodically to make sure that no items are expired. If an ingredient is expired, it is thrown out and replaced. Items that are either damaged or seem unclean must also be thrown out.

  A significant part of running a restaurant is making sure that you always have enough stock. To manage this, the owner checks the inventory daily. If the Inventory is well stocked, the manager will take a look at it the next day. However, if the inventory is not well stocked, he must take note of any item or ingredient that is running low and contact his supplier to place an order. Once the order is received, the manager updates the inventory, and the cycle continues.
- (5 marks) Appendix 1 – Use cases: A system diagram for the entire information system.
- (12 marks) Appendix 2 – Filled out use case templates of at least three representative use cases found in the system diagram (4 marks each)

- (16 marks) Appendix 3 – UML diagrams: Activity and sequence diagrams for two use cases in Appendix 2 (four diagrams, 4 marks each),
- (6 marks) Appendix 4 – A class diagram for the entire existing information system.
- (4 marks) Appendix 5 – A state chart diagram for a significant class
- (4 marks) Appendix 6 – Copies of forms and other documents used by client. If there are very many, then only include samples. If there already is a computer system, include screen shots.
- (2 marks) References/Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting

4th Deliverable – User stories (55 marks)

During meetings with the client, you will use the techniques of writing user stories to elicit requirements for a **new implementation** of an information system for the business. These will document both what the client is doing now, as well as what the client wishes to be able to do.

As explained in class, you will work with the client to produce at least 35 user stories on 3 x 5 index cards, and small PostIt® notes. If you are working remotely there are a number of different tools and environments available; for example, Trello[5], Miro[6], Mural, Asana, Stories on Board[7], MS Planner (part of Office 365 – available from the College), or even MS Excel.

This deliverable should contain the following information:

- (3 marks) Front matter (see 1st deliverable)
- (4 marks) Executive Overview (see above for a description).
- (3 marks) Summary description of the client (Revised – indicate what has changed or has been added since the last deliverable)
- (5 marks) Business problem – what is the business problem that the new system will resolve. It may have changed now that you have a better understanding of what the client wishes to be able to do.
- (10 marks) Narrative description of the future information system – This summarizes what is described by the user stories in the appendices. It should read like a story about what the various users do when they interact with the system. Write it after compiling the user stories. Here are excerpts from a previous student description of the future information system:

  We have identified three primary roles relevant to the system. First there is the user as a generic role; it represents anybody using the system, including anybody attempting to use it without prior authentication as well as anybody trying to perform a non-privilege-based action such as changing a password or logging out. Then there is the owner who can be thought of as the system administrator; he has escalated privileges and can therefore control who else has access to the system as well as what privilege level they have. Finally, the cook is a role with limited write access who can only modify the quantities associated with existing items. The system will purposely maintain a separation between roles and permissions so that further roles may be added if the need arises.

  The actions an unprivileged user can take will be minimal. A user will be able to try and login and, should he provide the matching identifier and password, will be allowed to do so. A user will be allowed to change his own password provided he can supply his previous password as well as enter a new password matching security criterion twice. The user can of course also terminate a session to prevent unauthorized uses by others.

  Once authenticated, if a user turns out to belong to the owner role, he officially becomes an owner. An owner can create an unlimited number of users to whom he can assign any existing roles (currently either owner or cook). The owner has create and delete permissions for items and categories as well as the categorize permission for items. This means that he can add and remove items from the inventory and likewise manage categories to which items may belong and freely assign items to said categories.

  One key ability for the owner is to quickly see what stocks are available at a given time; to that end, the system will provide a number of facilities. The owner will be able to sort items by alphabetical order as well as stock quantity; alternatively, an overview by categories will provide an at-a-glance summary. The owner will also be able to set a

[5] https://www.youtube.com/watch?v=_iodOh-QTww
[6] https://miro.com/templates/user-story-map/
[7] https://storiesonboard.com/storymapexamples.html

reorder threshold on given items so that the system may advise him of short supplies automatically. This alert will be displayed prominently upon login, but it is also possible that e-mail or other forms of notifications might be dispatched. …

The owner will have the ability to assign a reorder quantity to individual items so that a predefined number is available when stocks run low and the time has come to order again. A list will then be generated with item names, associated suppliers and quantities attached. When reordering is completed, the owner will then be able to mark an item as being on order, set the date upon which delivery is expected as well as take any additional notes which may be relevant. This information will then be visible within the inventory listing until the order is marked as delivered.

The system may also allow the owner to add and schedule events. …

The cook, by comparison, has much more limited access to the system. He can use the various methods described above to find out stock levels associated with certain items and update those values as needed. The cook will also receive the same low stock warnings as the owner ….

- (10 marks) Appendix 1 – Describe the process you used to obtain the user stories (5 marks), followed by a list of at least 35 user stories – print out the detailed contents of the Product Backlog (5 marks).
- (5 marks) Appendix 2 – At least two user story tests for each user story in Appendix 1 and transcriptions of all user story tests. These will form the basis of your Acceptance Tests.
- (10 marks) Appendix 3 – User story map – Explain your choice of tool for representing the story map (5 marks). Printout of the Story Map (5 marks). Make sure to include references to your list of user stories. If you are not including all the stories in Appendix 1, or if they are different, explain why.
- (2 marks) References/Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting

5th Deliverable – Prototype UI and client comments (80 marks)

Having done an initial analysis, you will now develop a prototype **user interface**. You may have even started this exercise during your first meetings with the client.

Ideally, the first prototype user interface will be hand-drawn on paper using a pencil or pen, while in the presence of the client. One of the reasons for using this basic technology is that it is not intimidating, and it encourages the client to have a hands-on involvement with the project, right from day one. The client will be able to provide immediate feedback. Also, since the initial prototype is "only" on paper, it is really easy to change it and even throw it away and start a new one [It is possible that you have totally misunderstood what the client's business is all about].

If it is not possible to do this in person, you can use a shared drawing space such as Google Drawings or MS OneNote. What is important is that you and your client collaborate in developing an interface.

As part of the UI design process, you will establish usability guidelines to be followed. Normally, following these guidelines will result in an interface that consists of a number of interactive forms or screens that do not normally present the contents of the data base as a table. *Exceptions to this must be approved by the instructor*.

Based on the client's comments on this first prototype, you will produce a second computer-drawn prototype, using drawing tools, Photoshop, or HTML and CSS. Note that this prototype is purely visual; that is, there are no actions taking place. The idea is to give the client some idea of what the implementation will actually look like. There should be several screens (at least 5) to be presented to the client, with an indication of the flow of interaction between them.

Based on your improved understanding of the client's needs you will now update, and very likely add, remove, and change, the user stories you developed for the fourth deliverable. Your story map will also have changed.

This deliverable should contain the following information:

- (3 marks) Front matter (see 1st deliverable)
- (4 marks) Executive Overview (see above for a description).
- (5 marks) Revised summary narrative description of the project; indicate what has changed or has been added since the last deliverable.
- (4 marks) Business problem – what is the business problem that the new system will resolve. It may have changed now that you have a better understanding of what the client wishes to be able to do.
- (10 marks) List of at least 10 usability guidelines being followed, with brief explanations. Include the source of each guideline, complete with reference.
- (10 marks) Copies of the prototype interfaces you developed with the client. There must be **at least 5 different screens** in this prototype.
- (19 marks) Client's comments
  - (5 marks) Describe the process used to interact with the client (in person, via Zoom, using email, etc.?), and
  - (4 marks) the client's comments on the first (hand-drawn) prototype, and
  - (10 marks) the client's comments on the second (computer-drawn) prototype.
- (10 marks) Describe the changes from one prototype to the next: For example, "*In the initial prototype, the login button was at the lower right corner of the screen; the client wanted it to be more prominent, so we placed it in the center of the screen, made the text larger and the button itself bright green*."
- (5 marks) Appendix 1 – Revised User stories and tests – indicate what has changed, plus any new user stories.
- (5 marks) Appendix 2 –Revised story map – indicate what has changed.
- (2 marks) References/Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting

6th Deliverable – Database design (73 marks)

Based on your understanding of the client's requirements, you will design a relational database system. In order to describe this design, referring to the initial class diagram (3rd deliverable), you will produce an ER diagram and describe the relational table structure. By examining the user stories, you will discover the entities and their relationships.

You will also determine the physical characteristics of the database system. You are free to use Oracle, or an implementation of SQL for this implementation, taking into account the hardware requirements of each DBMS, as well as your ability to integrate it with the rest of your implementation. You will explain the technical specifications for storing and retrieving data. The goal is to create a design for storing data that will provide adequate performance and ensure database integrity, security and recoverability. It is at this point that you need to consider the size of the database, both now and for the future (three to five years). You also need to think about how quickly the database needs to be accessed.

Your deliverable should include the following:

- (3 marks) Front matter (see 1st deliverable)
- (4 marks) Executive Overview (see above for a description).
- (3 marks) Business problem – what is the business problem that the new system will resolve. It may have changed now that you have a better understanding of what the client wishes to be able to do.
- (10 marks) Narrative description of the database design. Here are three different examples from previous students' deliverables:

  > *… We have 4 actors taking part in this system. They are the admin, students, teachers and librarians. Each actor has their respective task, differentiating the importance of each of them. For example, the admin is able to look at how many times a particular student logs in to book a room, and they can see what times are at peak for study room bookings. A regular student would not have access to that type of information, and therefore that would not be one of their tasks. …*
  >
  > *… When the Marketing Director wants to add a new client's information, he has to add the client's Contact number, advertisement ID, the business name, the full name of the person to contact in that business, the address, the telephone number, fax number, email and the day the client was contacted. The Marketing Director or the Secretary can also view and update the existing clients' information stored in the database. The users can also add new advertisement information on their new clients such as which business sector or section they want to be, edition, size of the advertisement and the city they are in. The users can also view and update the existing advertisement info on existing clients in their database. When the users choose to view the advertisement, the cost of the advertisement is also displayed and the contract information. …*
  >
  > *There are two actors in total, which are the manager and his employees. The manager can view, edit, add or remove employees from the Employees database, he can view, edit, add or remove clients from the Clients database, he can add, remove or edit appointments from the Appointments database, he can view or edit the schedule from the Schedule database, he can view Projects from the Projects database and he can add, view or edit supplies from the Supplies database. Meanwhile, the employees can only view supplies from the Supplies database and view the schedule from the Schedule database. This database design prevents important information from being inappropriately meddled with by unknowledgeable staff, which is why the employees' access to the database is restricted as such. …*

- (5 marks) Include a block diagram showing how each user uses the application to interact with the various parts of the database. Show the interaction between the user and the various tables. Something like the following (*Figure 2*), with appropriate labels:

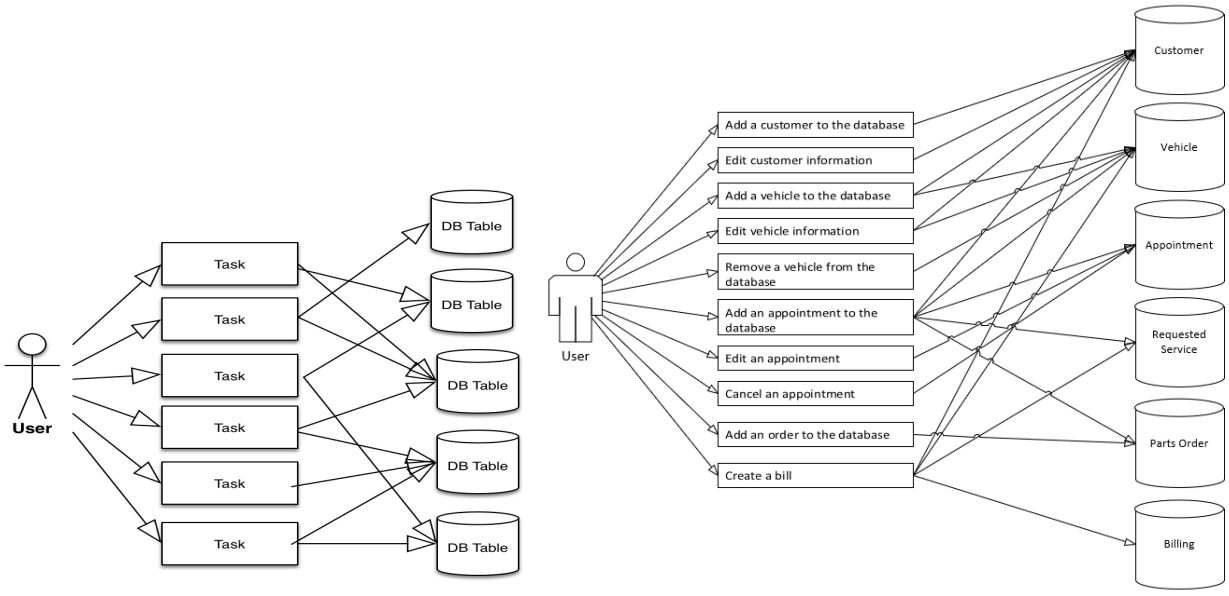


Figure 2: Example block diagram

- (10 marks) Appendix 1 – Create a data dictionary, complete with entities, relations and attribute types.
- (15 marks) Appendix 2 – Using MS Visio, produce an ER diagram of the database supporting the information system, complete with cardinalities (5 marks). Reproduce the class diagram from the 3rd deliverable and explain how and why your ER diagram is both similar to, and different from, the class diagram in Appendix 5 of the 3rd deliverable (10 marks).
- (4 marks) Appendix 3 – Descriptions and explanations of
  - (2 marks) Indexes and the database architecture of your design. What indexes are you going to be using in which tables, and why?
  - (2 marks) Query optimization in your design. There are going to be many queries. Do you need to optimize them? If so, why and how? If not, why not?
- (9 marks) Appendix 4 – The projected size of the database (in MB or GB), now and for the next three to five years. For each table, determine the maximum size of each record. Then, estimate the maximum number of records per table, and then come up with a value for the maximum size of the table. This will give you the maximum size of the database now. Make assumptions about how the number of entries in the database will grow, and then use that number to determine the future size of the database. Show the detailed calculations and assumptions made to arrive at your estimate.
- (5 marks) Appendix 5 – Explain the access speed required, and how your design will permit this. How often will the database be accessed? How much data will need to be stored or retrieved? What kind of response time will be necessary?
- (2 marks) References/Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting

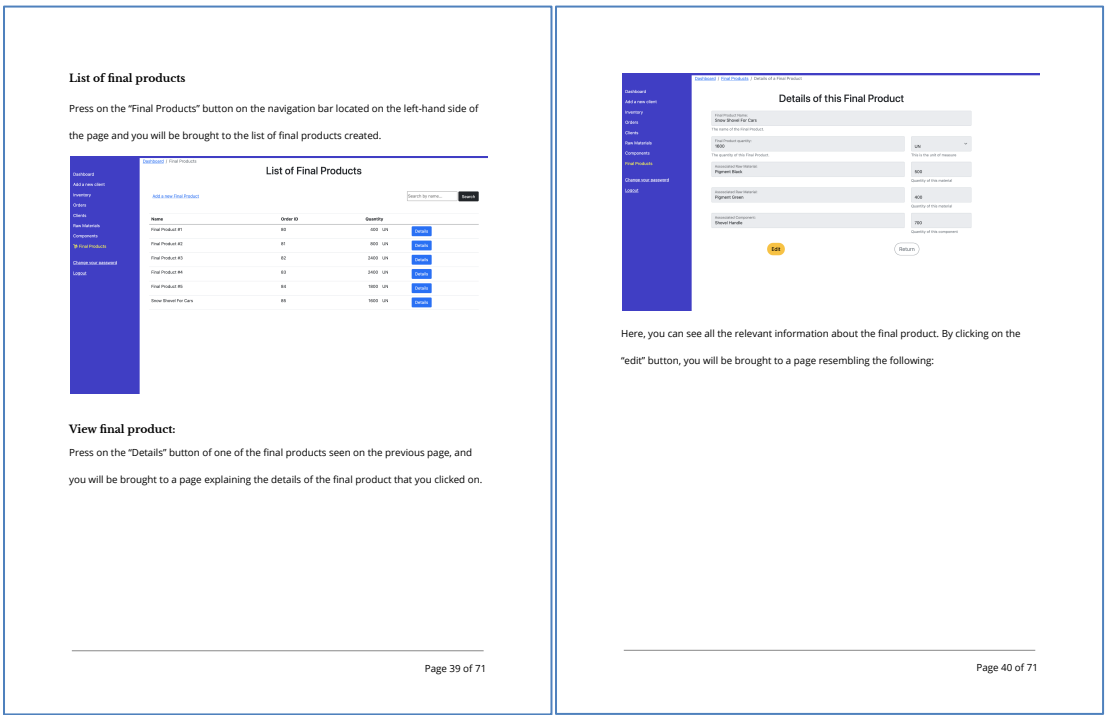
7th Deliverable – Implementation and client comments (76 marks)

After you have taken into account the various comments of the client to the previous prototypes, you will present the client with an implementation. **In consultation with the instructor**, you will determine which parts of your database design (that is, which tables and which functions) you will implement. Your implementation must allow the user to do at least the following (note that some of these actions are required by the instructor):

- Perform input error checking.
- Search for an item, or a group of items, in the database.
- Add data to the database.
- Retrieve data from the database and display it on screen and print it directly to a printer.
- Change (update) data in the database.
- Delete data in the database.
- Produce at least one nicely formatted summary report of the database contents.
- As per Deliverable 5 there will be at least five screens in the user interface.

As part of the implementation, you will provide the client with a user guide. This guide explains in detail how the different users can use the system to accomplish their tasks.

Here is an example of part of a User Guide:



List of final products

Press on the "Final Products" button on the navigation bar located on the left-hand side of the page and you will be brought to the list of final products created.

List of Final Products

View final product:

Press on the "Details" button of one of the final products seen on the previous page, and you will be brought to a page explaining the details of the final product that you clicked on.

Page 39 of 71

Details of this Final Product

Here, you can see all the relevant information about the final product. By clicking on the "edit" button, you will be brought to a page resembling the following:

Page 40 of 71

As you present the implementation to the client, you will collect the client's reactions and comments on the implementation. These comments will form part of the present deliverable.

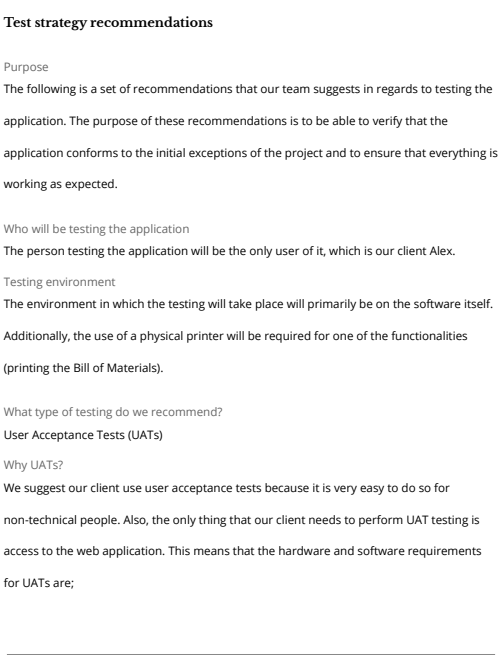
This deliverable should contain the following information:

- (3 marks) Front matter (see 1st deliverable)
- (4 marks) Executive Overview (see above for a description).
- (5 marks) Summary description of the client, including the final business problem addressed.
- (3 marks) Revised summary narrative description of the system.
- (5 marks) Client's comments
- (5 marks) Discuss the various decisions you made regarding both the **design and implementation** of the system, including both visual design and software design, including choice of programming environment. For example (from previous student work),

... For the design and implementation of the system, we had decided to keep the functionality and looks consistent with the prototypes that we had previously created. As the client had designed the look of the database, we wanted to satisfy his requests while also providing the functionality that he wanted.

In regard to the design aspect of the system, there were some slight changes we had to make due to the limitations of C# programming. In C#, if you are creating a form application which is what our database is using, there will be buttons at the top of the screen in the form of icons. By pressing the icons, they perform various tasks such as deleting, creating, saving, etc. Even though this is not what we initially planned to have in the system, we decided to continue using C# as this is a minor inconvenience for the overall aesthetic of the application. ...

- (5 marks) Description of current security measures
- (16 marks) Future work (what needs to be done to complete the project). Include pictures of the story map to show what was done and what was not done. Note that what was removed from the system entirely should not appear in the story map.
  - (3 marks) User interface improvements
  - (3 marks) Unimplemented user stories and functions
  - (5 marks) Recommendations regarding future security measures to be taken.
  - (2 marks) Recommendations regarding unit and integration test strategies to be used.
  - (3 marks) An acceptance test plan. Here is an example:



Test strategy recommendations

Purpose

The following is a set of recommendations that our team suggests in regards to testing the application. The purpose of these recommendations is to be able to verify that the application conforms to the initial exceptions of the project and to ensure that everything is working as expected.

Who will be testing the application

The person testing the application will be the only user of it, which is our client Alex.

Testing environment

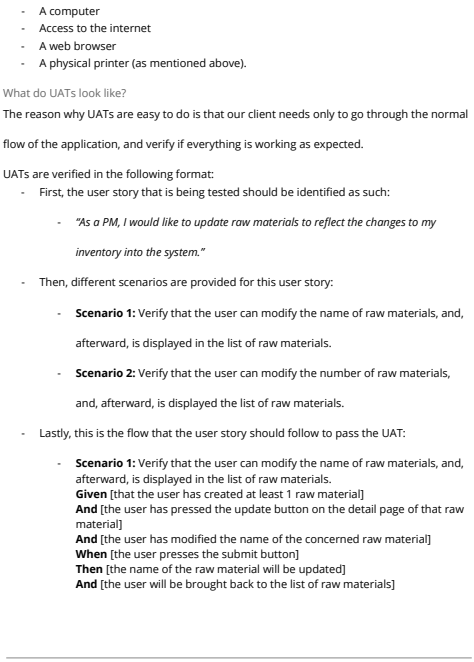
The environment in which the testing will take place will primarily be on the software itself. Additionally, the use of a physical printer will be required for one of the functionalities (printing the Bill of Materials).

What type of testing do we recommend?

User Acceptance Tests (UATs)

Why UATs?

We suggest our client use user acceptance tests because it is very easy to do so for non-technical people. Also, the only thing that our client needs to perform UAT testing is access to the web application. This means that the hardware and software requirements for UATs are;



- A computer
- Access to the internet
- A web browser
- A physical printer (as mentioned above).

What do UATs look like?

The reason why UATs are easy to do is that our client needs only to go through the normal flow of the application, and verify if everything is working as expected.

UATs are verified in the following format:

- First, the user story that is being tested should be identified as such:
  - *"As a PM, I would like to update raw materials to reflect the changes to my inventory into the system."*
- Then, different scenarios are provided for this user story:
  - **Scenario 1:** Verify that the user can modify the name of raw materials, and, afterward, is displayed in the list of raw materials.
  - **Scenario 2:** Verify that the user can modify the number of raw materials, and, afterward, is displayed the list of raw materials.
- Lastly, this is the flow that the user story should follow to pass the UAT:
  - **Scenario 1:** Verify that the user can modify the name of raw materials, and, afterward, is displayed in the list of raw materials.
    Given [that the user has created at least 1 raw material]
    And [the user has pressed the update button on the detail page of that raw material]
    And [the user has modified the name of the concerned raw material]
    When [the user presses the submit button]
    Then [the name of the raw material will be updated]
    And [the user will be brought back to the list of raw materials]

- (5 marks) Appendix 1 – Revised user interface – indicate what changed from the prototype to the implementation, and why. Use printed screen images to highlight the changes.
- (10 marks) Appendix 2 – Printed version of the user guide. This should be written from the user's point of view; that is, start with the task that the user wishes to perform, and then explain how to do it.
- (10 marks) Appendix 3 – List with screen images showing which parts of the interface implement which User stories.
- (2 marks) References/Bibliography/Works cited (APA Style)
- (3 marks) Spelling, grammar and formatting

LIA Prototype Implementation

The implementation will be evaluated as part of the Learning Integration Assessment, as described in the course outline:

LIA Prototype implementation (14%)

- Accurate analysis of client requests and requirements (2%)
- Accurate analysis of the features of the computer equipment and applications (2%)
- Choice of application development standards, methods, and best practices (2%)
- Assessment of the software and hardware components to be used (2%)
- Appropriateness of the design, solution, and implementation techniques (2%)
- Compliance with application development standards, methods, and best practices (2%)
- Accurate drafting of unit, integration, functional, or acceptance test plans (2%)

If you are not present for the demonstrations scheduled for your team and for the other teams in your section, you will receive a mark of 0 for the implementation.

The instructor will attempt to use the prototype, posing questions to each member of the team, to verify their understanding of both the implementation and the various design decisions that were taken to produce the implementation. This will be done either from a USB drive, a downloaded .zip file (using, for example, WeTransfer.com), a web site, or a shared screen in a Teams meeting.

Use the MVC architecture for your implementation.

Show your file structure to the instructor and make the source code/web site available to the instructor either on a USB drive, or in a downloadable format so that a copy can be transferred to the instructor.

The implementation must:

- Have at least five records per table installed in the database.
- Have at least five screens as part of the user interface.
- Perform input error checking.
- Conform to well-established UI guidelines.
- Allow the user to
    - Search for an item, or a group of items, in the database.
    - Add data to the database.
    - Retrieve data from the database and display it on screen and print it directly to a printer.
    - Change or update data in the database.
    - Delete data in the database.
    - Produce at least one nicely formatted summary report of the database contents.

Work Cited

Information silo. (n.d.). Retrieved December 3, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Information_silo