Business case

The introduction starts with describing the purpose of this document and what it will cover.

**1.1 The problem**

The heading above is just a placeholder. You need to provide a heading which draws attention to your actual problem, e.g. “Healthcare worker patient notes”. This section should provide a general description of the problem you are addressing. Include information such as:

* What the problem is – not the lack of software, but what is the actual real-world problem that you can help solve by providing software.
* The domain of the problem (i.e. where it takes place)
* Who is encountering the problem?

Be as descriptive as you can. Think about how much information you will provide a possible financial backer to make them care about the problem and make an investment.

**1.2 Project objectives**

In this section you should list the real-world problem issues that your software intervention will address, e.g.:

* Replace the paper-based note-taking system used by healthcare workers with a digital alternative.
* Ensure that notes cannot get lost, by providing an off-line saving system.
* Provide a central means by which to search through historic notes.

Think of these as the main things that your system will do to address the real-world issue.

**1.3 Problem background**

This is the literature review section of the document. In order for reader to better understand the problem, this section must provide an overview of the problem area using existing sources, e.g. web references, conference papers, white papers, patents or journal articles. You are required to use proper APA referencing. The idea with this section is that if a reader has no knowledge of the real-world problem area; this section should provide them enough background to understand the issue, e.g. an overview of the process that healthcare workers use when visiting patients in remote areas.

**1.4 Related systems analysis**

You are required to find (at least) 3 software-based systems which address the same issue as the one addressed by your proposed system or systems which use a similar process to what you envision for your system.

For each related system you are required to provide the following:

* The name of the system
* Platform (e.g. web, Windows / Mac, Android / IOS, etc)
* Description of the system
* At least one screenshot. All screenshots must have properly numbered and worded captions. The text in your document must explicitly refer to it, e.g. Figure 1.1 demonstrates the login screen of System X.
* A list of features / processes that you would like to adapt and incorporate into your system (i.e. the reasons why this system caught your eye)
* A list of features / processes that you would like to avoid in your system.
* Each related system should be properly referenced.



[This Photo](http://clasticdetritus.com/2008/11/10/wild-animals-in-the-field/) by Unknown Author is licensed under [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/3.0/)

Figure 1.1 System X Login Screen

**1.5 The project plan**

A project, or project management plan is a document that contains a project scope and objective. Typically, it will be represented as a Gannt chart to make it easy to convey information.

A project plan should answer these questions:

* What are we delivering?
* How will we deliver on time?
* Who is working on the project and in what role?
* What miles stones/goals are set for project?

Before you can start your project plan you need to understand (at a minimum) the following things:

* What does your client expect/need?
* The goals of the project
* What is the decision-making process of your client, and how will they approve and review the project work?
* Who is the sponsor?
* Who is the project manager?
* What other stakeholders are important?

As project manager you also need clarity from your client to know exactly what their expectations are. Sometimes this requires you to ask some hard questions to get a clear understanding.

Questions that may impact a project plan:

* How will you collect feedback?
* Who has the final sign-off?
* What is the project deadline?
* What is the availability of the project team?
* How often will the teams meet for feedback?
* Does your team have a history of successful projects?
* What can prevent the project from being a success?
* What tools and methods of communication will the team use?

*Your project plan should include the following information (at the very minimum):*

* *Project and client name (make one up)*
* *Delivery date and version*
* *Milestones and deliverables*
* *Clear task durations with start and end dates (using a Gantt chart)*
* *Dependencies for tasks (i.e., should anything happen before this task can happen).*

**1.6 Risk Analysis**

A risk can be defined as an unexpected event or condition, which can impact negatively on the entire project. Some will argue that the event or condition can also affect the project in a positive way.

There are many models, which are used to identify different risks. But we will not go into specific ones here. In general, we can say that to categorise risks can help in the process. You will quickly discover that you need to limit yourself regarding risks during a brainstorming activity. Otherwise, the list of events and conditions can become very long. You need to assess how likely it is that the particular event occurs. For instance: *The entire project could be ruined if a woman in labour runs across a road and a driver has to swerve to avoid her. The driver crashes into a bus and loses a wheel, which rolls across the road, pulls along a bag of dynamite, which was about to be used. The dynamite then explodes next to a lamppost, which shoots, like a rocket, through the window of the project teams office. The rocket hits the project leader in the head and the project leader, by mistake, deletes all the code, which has been written. The rocket continues through the office and finally lands on a shelf filled with super magnets, which then fall onto the server, which contains all the backups and removes them all.* But how likely is this?

Once you have identified the risks, you need to decide how you will handle those risks. It is a good idea to divide the risks into four main categories:

**Acceptable risks:** We decide that the risk is very unlikely, or possibly, that the costs if that event happens, are lower than the expenses which are necessary to prevent the event from taking place.

**Avoidable risks:** We take action which ensures that we prevent a certain event or condition. We might create backups of all the code and store it in two separate locations. Then we avoid the risk of losing anything,

**Minimizable risks:** We take steps to ensure that the likelihood of something happening is less. We can create a bonus structure to make sure that people do not leave.

**Transferable risks:** This means that we transfer the risk to someone else. For example, we might send part of the work to another company. Then it is the other company, which needs to assume the risk. Once we have categorised all the risks, we need to decide exactly what needs to be done, if something should be done. We need to prepare ourselves as best we can so that any potential events have as little impact on the project as possible. We can, of course, have free fruit available at work and encourage everyone to exercise regularly. But someone will get sick in any case and then it is great to know how we handle that. Who is going to take over the sick person's work? When is it necessary for someone to step in? Should the person step in be someone in the team or someone from outside the team? In the case of the latter, where should the person be brought in from? How much will this cost?

*Identify (at least) 4 risks to your project. If you have time, add more, since the more you are able to identify at the start, the better you can plan for things to go wrong (and they usually do). For each risk, state which type of risk it is and how you will treat/handle the risk.*

**1.7 References**

Your reference list here.