Report

# Table of Contents

Table of Contents 2

Introduction 2

Problem Definition 2

Main Section 3

Company Description 3

Stakeholder Analysis 3

Feasibility Studies 7

Technical Feasibility Study 7

Schedule Feasibility Study 9

General Analysis 11

SWOT 11

Summary 12

Bibliography 12

Appendices 12

# Introduction

## Problem Definition

Our customer, App Academy Aps, wants a program with following specifications:

* Standalone, single executable file.
* Generate a course certificate, which holds a course name, course participant's name and the date. This shall be saved as a jpg file.
* The certificates must have a unique IDs.
* The certificates shall be sent by e-mail with customizable text.

Furthermore, the client has wished for features which aren't essential for the program, but would be nice to have:

* “Drag 'n' Drop” functionality for generating a certificate template.
* Upload course material.

Due to close partnership with App Academy Aps we haven't been forced to set restrictions.

Our initial thoughts for solving this are to first focus on the design part, then hopefully construct the product without much difficulty. Also we have decided to describe App Academy Aps' business structure.

# Main Section

## Company Description

## Stakeholder Analysis

**Stakeholders:** KEA, Testers (Class Mates), Project Group, App Academy(Kristian), Course Responsibles

PLACEHOLDER FOR FIG. 12.7 I ORGANISATION 5. UDGAVE

Grey Eminence: KEA

Hostage/ResourceStakeholder: Kristian

Hostage: Course Responsibles

ResourceStakeholder: Project Group

External Stakeholder: Testers

**Grey Eminence: KEA**

**KEA as a Stakeholder:**

We’ve decided that KEA has the role of a Grey Eminence. We’re not going to pay a lot of attention to KEA as an institution during our project, but it has the power to change the entirety of the project if it wants and sees an interest in it.

**KEA’s attitude towards the project:**

KEA has no direct wish for the actual program to be completed, but wishes the project group to succeed as the project group is part of their education system.

**KEA’s potential for conflict:**

KEA has accepted the terms of the project as an exam project and does overruled any saying Kristian might have regarding the continued direction of the project.

**KEA’s resources:**

They can provide help with all aspects as they provide teachings in all subject components needed to deploy the system. Study rooms along with projectors and whiteboards have all been provided by KEA.

**Hostage/ResourceStakeholder: Kristian, App Academy**

**App Academy as a Stakeholder:**

Kristian fits multiple roles he’s both a hostage and a resource stakeholder simultaneously. We’ve also argued that during the inception of our project, he was a Grey Eminence, dictating the terms and requirements for the project. He no longer fits that role due to the stage the project’s at. He is a hostage in term’s of him not being an active participant in the elaboration or construction phase. Yet as a client, he will be immensely important during our transition phase, and thus he’s simultaneously a Resource Stakeholder.

**AppAcademy’s attitude towards the project:**

AppAcademy has a low activity in regards to the project, to them it’s a nice to have program. Not a core business requirement. Their motive is facilitation of administrating part of their core business. They are contributing with running reviews and certain design aspects.

**AppAcademy’s potential for conflict:**

AppAcademy has a high potential for conflict with KEA as they have to adjust their expectancies to the boundaries set by KEA.

**AppAcademy’s resources:**

They are not providing any resources, as the software will be running as a standalone client.

**Hostage: Course Responsibles’**

**Course Responsibles’ as Stakeholders:**

These are the employee’s at App Academy that hold courses. They have no influence on any parts of the project and will have to accept the final product the way it is.

**Course Responsibles’ attitude towards the project:**

Their attitude could be negative as the project will impact their usual workflow. The project risks getting derailed due to their unwillingness to use the software.

**Course Responsibles’ potential for conflict:**

There is a chance for conflict depending on how AppAcademy introduces it to their employees. The conflict not be with the Project Group though.

**Course Responsibles’ resources:**

They have the potential to provide precious feedback, which could be useful.

**Resource Stakeholder: Project Group**

**Project Group as a Stakeholder:**

We’re present in all 4 UP Phases: Inception, Elaboration, Construction, Transition. We’re within certain limits defined by our Grey Eminence, in charge of the entirety of the project as the success or failure of it, falls upon us.

**Project Group’s attitude towards the project:**

Our attitude is that the project has ultimate importance. We will be the primary driving force in seeing that the project becomes realized and deployed. Our motive for this is both in learning and educational progress and we will thus be dedicating all of our time towards this.

**Project Group’s potential for conflict:**

There is a great risk for potential conflict with most of the other Stakeholders. KEA will hold great power over our project and as such, these kind of relations can become quite strenuous if not tackled in a constructive way. It is thus imperative that the Project Group maintains a positive attitude towards KEA and external influence in general.

**Project Group’s resources:**

We will be pulling from our knowledge in Software Development in order to create the project. We will be contributing most of our time and personal computers to see it realised.

**External Stakeholders: Testers**

**Testers as Stakeholders:**

These are our class mates, that have agreed to help us with testing the product. At first glance they don’t hold any influence on our project, but depending on feedback from them, they could become very important all of a sudden, and their influence upon our direction for the project could dramatically increase.

**Testers’ attitude towards project:**

Generally positive and their main contribution will come from their feedback as fellow students testing our software.

**Testers’ potential for conflict:**

Competitiveness could be an issue, but at this stage there has been no sign thereof. Again it will be important that the Project Group remains positive even if harsh feedback is given.

**Testers’ resources:**

Per se, the Testers will be contributing with their time as a Resource.

**Risk Assessment[[1]](#footnote-1)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Risk | Probability | Consequence | Product | Preventive Measures | Responsible | Solutions | Responsible |
| Illness | 2 | 1 | 2 | There is not much we can do here | Project Dev’s | The rest of the team will have to compensate for the missing know-how. | The group |
| Lack of testing | 1 | 7 | 7 | Continously test so that it’s constantly maintained | The group | Implementing a rule that every method will have to be tested as it’s coded. | The group |
| Implementaiton | 1 | 10 | 10 | Maintain a dialogue with Kristian | Kristian | Make sure we document our code and user manual correctly. | The group |
| Resources Missing | 4 | 3 | 12 | Plan ahead. | The group | Have more than 1 plan to a problem | The group |
| Bad Estimates | 5 | 7 | 35 | Maintain our Estimates after each meeting/session. | The group | Adapt to changing circumstances, be dynamic. | The group |

**Risk Analysis Comment**[[2]](#footnote-2)

**Illness**: Due to the nature of how we’re working on this project, there is a very minimal chance of Illness impacted since we’ve pooled our resources, it basically means that we would all have to fall sick simultaneously for illness to severely impact us. It is highly unlikely that this will happen.

**Lack of testing**: It is very important that we continuously test our software, so that we’re sure that everything works as intended. To minimize risk here, we will be looking to unit-testing as a tool.

**Implementation**: It will be hard to do much during implementation other than introduce Kristian to our product in the best way imaginable. Since he can code, this means that we should comment and document our code adequately so that he can trouble-shoot himself. Secondly we will have to make sure our user-manual is well done, so that “Project Responsibles” can properly solve their issues by consulting it.

**Resources Missing**: In case of computers crashing as an example it’s important that we have backups of our code. We’ve decided to use GitHub to manage the versions of our software and for live backup we’re using Dropbox. This way our code is located on several different systems and maintained there, preventing irrecoverable system crashes.

**Bad Estimates**: Considering that we’re still students with a limited knowledge and experience, it’s hard for us to “guess-timate” in a correct manner. Therefor it’s imperative that we meticulously adjust and maintain our budgeted time schedule. If things need to be speeded up we have to be prepared to do so in order to deploy a good product.

## Feasibility Studies

### Technical Feasibility Study[[3]](#footnote-3)[[4]](#footnote-4)

* **Hardware**
  + CPUS: Intel 64-bit x86 Arch.
  + Computers: 4 Student Laptops
* **Software**
  + Java: JDK, JRE
  + Operation Systems: Windows 7-8-10, Linux Debian Based, Mac OSX
  + Database: MySQL
  + IDEs: JetBrains IntelliJ – DataGrip, Workbench
* **Manpower**
  + Developers: Emil, Jonas, Dennis, Fredrik
  + Testers: Class Members
  + Reviewers: Kristian (Client)
  + GUI Specialist: Dennis
  + Extra Curricular Skills: C, SML, Python, Ruby, PHP, HTML 5

**Risk Assessment**

* Failure to attain expected benefits from the project: Not a Risk due to the circumstance that we are Students and any learning process is beneficial, to an extent.
* Cost Estimates: We will have no costs to execute this project, material needed is available at no charge from our school.
* System Performance Levels: There are no specific performance requirements, but of course we don’t want an unstable system.
* Integration: There will be no software/hardware risk since the software will be system independent[[5]](#footnote-5).

**Team Thoughts concerning Technical Feasibility**

We have taken into consideration hardware and software requirements and have come to the conclusion that there won’t be any issues regarding these. In terms of abilities, we feel we are adequately equipped to implement all features for this project. However we have a small concern regarding our planned implementation of the “Course Certificate Generator” in terms of feasibility, but feel that this is a minor issue and no hinderance for us to proceed with the project. We have other solutions ready if needed.

### Schedule Feasibility Study[[6]](#footnote-6)[[7]](#footnote-7)

We’re going to Budget our Schedule by breaking down the hours available to us.

Dates where we will be working on the project:

* April: 19, 22, 26, 29 / May: 3, 6, 10, 13, 17, 20, 24, 27, 31[[8]](#footnote-8)

We count 10 hours per date, which amounts to 130 hours per Developer and 520 man hours total.

These dates are extra curricular and don’t take into account the hours that will be gathered from class schedule.

There is a remaining 6.5 Weeks worth of class that will be spend on this project. 1 week worth of class hours is set to 17 hours. This amounts to 110.5 per Developer and totals to 442 man hours. This gives us a budget of 962 man hours and just 240 hours per Developer. This budget doesn’t take into account sick leave nor breaks.

We’ve set sick days a bit lower than national private sector average of 6.4[[9]](#footnote-9) to 5, because the project is scheduled outside flu season. The normal working hours for an employee is 37 hours per week, 7.4 hours per day. There is a total of 5 weeks vacation as a minimum which equates to 47 working weeks per year. Total amount of working hours per year is 1739, which translates to 235 days.

We now deduct 5 sick days from 235 total working days, which equates to 2.12%. Now apply 2.12% to our total developer hours of 240.5 which means we roughly get 1 sick day per developer during the course of the project, this results in a budget reduction of 4, total man hours is down to 958.

As a remainder we will also have to take breaks into account. We’ve set our preliminary limit at 1 per 4 hours of development time. 20% of off our budget of 958 equals 766. This means that we have a total of 191.5 hours dedicated to break time. Our budget is now reduced to 766 man hours, effectively 191.5 per developer, this number is real estimated work hours.

**Budgeted hours:**

|  |  |
| --- | --- |
| Segment | Developer Hours: 191.5 |
| ITO | 20 |
| SWD | 70 |
| SWK | 30 |
| OSCA | 10 |
| Budget | 61.5 |

We have decided to distribute 130 hours on our 4 primary project groups ITO, SWD, SWK and OSCA. This has left us with 61.5 hours as a buffer that can be applied where necessary. We think this is good since we’ve thus taken roughly 35% of our total developer hours and dedicated those to unforeseen events. We don’t think it would’ve been a prudent choice to plan with a 100% of available hours.

**Team Thoughts regarding Schedule Feasibility**

We decided to take a rather stringent approach to our schedule, seeing that making our total resources visible to us through numbers and facts would make it easier for us grasp and plan ahead.

Currency, finances, those were both reoccurring thoughts here, we decided to translate that into man hours, since this is the only resource that we will have to take into account. Another thing to remember here is that, we’ve decided to do the entirety of this project congruently as 4. Thus our focus is on our total developer hours of 191.5 and not the 766 man hours. This could be rather misleading if one was not aware. This decision means that we will limit our primary resource: time. We feel though as the development of this project is both a feat in productivity it is also a learning progress. By combining our abilities and knowledge we will gain a lot through shared knowledge and learning speed. Last but no least; faster productivity by combining efforts towards same goals constantly. Not one project member will be in the dark about any part of the project. Schedule wise, the project seems feasible at this stage, even with the consolidation of our man hours.

## General Analysis

### SWOT

## Database

# Summary

- Reminder: Husk at evaluerer hvordan vi glemte at implementere “Send certifikat” delen samt at Kristian også glemte at det skulle være med, da vi viste ham prototype 1.

# Bibliography

# Appendices

1. projekt\_og\_analyseredskaber.pdf / Figur 4.6 [↑](#footnote-ref-1)
2. Why Projects Succeed proactive risk strategi / Inspiration [↑](#footnote-ref-2)
3. Course Material: Materiale\_Feasibility\_Study\_copy, Feasibility\_Studies\_IT\_af\_Steve\_McConnell, Feasibility Studies Hoffer 8 sider [↑](#footnote-ref-3)
4. Appendices: 1, 2 [↑](#footnote-ref-4)
5. <https://docs.oracle.com/javase/8/docs/technotes/guides/#javavm> / Visited 19.04.2016 [↑](#footnote-ref-5)
6. Appendices: 3, 4 [↑](#footnote-ref-6)
7. Course Material: Materiale\_Feasibility\_Study\_copy, Feasibility\_Studies\_IT\_af\_Steve\_McConnell, Feasibility Studies Hoffer 8 sider [↑](#footnote-ref-7)
8. Captains Log [↑](#footnote-ref-8)
9. <http://nyheder.tv2.dk/2014-12-19-tjek-her-laegger-du-dig-mere-syg-end-gennemsnittet> / Visited 19.04.16 [↑](#footnote-ref-9)