Timetabling System

Nabeel Mir: N07017476, Thomas Webb: N0682255, Tanaka Pisa: N0719344, Zainab Olawoyin: N0620500

Requirements Specification

2018

Contents

[Introduction 2](#_Toc530672481)

[The Problem Statement 2](#_Toc530672482)

[Existing Solutions 2](#_Toc530672483)

[Stakeholders 3](#_Toc530672484)

[Desired Solution 3](#_Toc530672485)

[Out of scope 3](#_Toc530672486)

[Scope and Constraints 4](#_Toc530672487)

[Scope 4](#_Toc530672488)

[Constraints: 4](#_Toc530672489)

[Feasibility Evaluation 5](#_Toc530672490)

[Technical Feasibility 5](#_Toc530672491)

[Schedule Feasibility 5](#_Toc530672492)

[**Tasks** 5](#_Toc530672493)

[Functional Requirements 6](#_Toc530672494)

[Must 6](#_Toc530672495)

[Should: 6](#_Toc530672496)

[Could: 7](#_Toc530672497)

[Non- functional requirements 7](#_Toc530672498)

[Must: 7](#_Toc530672499)

[Should: 7](#_Toc530672500)

[Could: 7](#_Toc530672501)

[Use case/ mis-use case 7](#_Toc530672502)

[Use Case Diagram 8](#_Toc530672503)

[References 10](#_Toc530672504)

[Appendices 11](#_Toc530672505)

[Appendix A: Context Diagram of System 11](#_Toc530672506)

[Appendix B: Current Solutions 12](#_Toc530672507)

[Appendix C: Effort Estimation 13](#_Toc530672508)

# Introduction

This documentation aims to discuss and ensure the requirements requested by the stakeholder are considered when developing a new solution. Utilising the requirements provided by the stakeholder will aid in defining the scope of the new solution.

# The Problem Statement

Higher education timetabling systems have been superseded by the advancement of technology, which have left both students and faculty with ever persistent issues around planning and scheduling (Babaei, Karimpour, & Hadidi, 2015). In addition, the primary stakeholder has highlighted a core problem of under populating or over populating rooms.

# Existing Solutions

The existing solutions are listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Features | Strengths | Weaknesses |
| PowerSchool | * Mass add requests. * Mass delete requests. * Schedule, search and select group of students. * Autofill student, course and teacher information. * Automatic generation of the above information. | * Tailored successfully to American students in High School. * The administrator is capable of handling requests by students and faculty. * PowerSchool allows the administrator to build a master schedule in response to constraints borne from various scenarios. * The administrator has a choice of scheduling by room, course or teacher. | * The interface of the scheduler is seemingly outdated. |
| NTU Timetable | * Students can view their scheduled events. * Updated in real-time. * Module code for each session is displayed. * Faculty allocated to the session is visible. * Can synchronise the timetable to external calendar. | * Students get updated about any changes to the schedule. * Constant access because the timetable is constantly displayed 24/7 * Students can view their timetable on any mobile device with the synchronised timetable on their calendar. * Single sign on approach gives convenience, as they can use the same credentials they use to enter NOW, to access their timetable. | * There have been situations in the past where there have been clashes in scheduled sessions for students. * There have been incidents where sessions within a module haven’t been assigned to students. |
| ILOG Scheduler | * Capacity Scheduling. * Flexibility Scheduling. * Extensible Scheduling. | * Tailored to businesses effectively. * Ability to compute scheduling solution. * Specialised algorithms and models for scheduling tasks and durations. * Useful for businesses during the planning stage of projects. | * The administrator is required to have background knowledge of coding to utilise ILOG Scheduler. |
| Vcita | * Managing and viewing the team's calendar. * Allocate faculty to services and appointments. * Customer cancellations, scheduling and rescheduling of appointments. | * Synchronisation of the users work schedule with their calendar on any device. | * Used for small businesses rather than faculty and students in academia. |
| I360 | * Forecasting of daily workloads. * Manipulation of scheduled tasks by the administrator. * Constraint handling and adjustments to schedule. | * Scheduler considers faculty availability and their skill set to schedule tasks. * Adjustments to faculty breaks can be made. * Data generated from faculty logged into the system, to forecast workload. | * To request changes, faculty must contact the administrator externally as opposed to within the software. Inconvenient for faculty. |

# Stakeholders

The identified stakeholders utilising the system are:

* Mr Nigel King (Administrator)
* Faculty
* Students

# Desired Solution

The project proposed by the stakeholder is to develop a timetabling system for students and faculty in higher education. The core aspects should generate the timetable without conflicting sessions by taking into consideration constraints and to provide a visual display of upcoming sessions of an individual’s schedule. The secondary requirements are ticket handling and change requests. There are existing solutions that provide the service of scheduling.

# Out of scope

The timetabling system prototype will not provide a solution for all courses and stakeholders in the university. Cross-platform integration will not be implemented. Daily and monthly views of the timetable will not be included in the user interface.

Other functionalities that will not be implanted are as follows:

Students could be redirected to a new email with a faculty members email in the 'To' field on Microsoft Outlook when this option is selected through a focused event.

Synchronisation of onsite facility events to the timetabling system.

Smart alarm for faculty and students, to alert them before a session starts. The user can select the timescale they wish to be alerted before the specified session.

Inclusion of document links to the module code within the timetabling interface.

# Scope and Constraints

## Scope

As shown below, by using the MoSCoW analysis method the requirements of the system have been separated into four categories (Must, Should, Could and Won’t).

The project will consider a building that contains 2 lecture theatres, 6 labs, 4 seminar rooms and 2 meeting rooms. There will be 10 members of faculty and 110 students across once course with 6 modules.

## Constraints:

|  |  |  |  |
| --- | --- | --- | --- |
| Constraint | Internal/External | Present/Future | Mandatory/Desirable |
| When product is launched a full database should be used to gain access to the module and room information | E | F | M |
| A prototype must be produced by 8th February | E | F | M |
| To prevent scheduling clashes within the timetabling system, we must build a timetable building algorithm that avoids these conflicts | I | P | M |
| A capacity optimization algorithm should be included within the timetabling system, to comply with health and safety regulations | I | P | D |
| To prevent excessive requests from a single user, we should allow the admin to approve unautomated processes within the timetabling system | I | P | M |
| To reduce the workload for the admin, we should build an automated system to approve some requests | I | P | D |
| The stakeholder would prefer that curriculum sessions are not scheduled outside of 9am to 5pm | I | P | M |
| when storing information about students and faculty, the system should comply to GDPR (2018) | I & E | P | M |

# Feasibility Evaluation

## Technical Feasibility

A new system will be created for the implementation of the timetabling system. The prototype for the system will be a Java based desktop application and will not be implemented on mobile devices.

The prototype will not be incorporated within a larger system. It would not be technically feasible to implement an embedded system; hence a distributed system will be used. The timetabling system would later be introduced as embedded and would function in a manner like that of Nottingham Trent University’s timetabling system (Nottingham Trent University, 2018).

The system will use a MySQL database which is a very fast, robust, relational database management system which controls access to the data and ensure multiple users can access the data concurrently (Welling & Thomson, 2003).

## Schedule Feasibility

## **Tasks**

The estimated delivery timescales were defined in consideration of the team’s commitments.

The tasks and milestones based upon the stakeholder's expectations of delivery timescales are below:

**Task 1** business context: 1st November

**Task 2** Feasibility Evaluation: 9th November

**Task 3** Scope and Constraint: 10th November

**Task 4** Use case/misuse case: 12th November

**Task 5** Use case estimation: 16th November

**Milestone: Submission of initial report & initiation of second: November 22nd**

**Task 6** Design: 6th December

**Task 7** Implementing functionality :6th January

**Task 8** testing: 12th January

**Task 9** Report Completion: 8th February

**Milestone: Report and Project Deadline: February 15th**

Four team members will be available throughout the duration of the project. The team will be working on the project approximately 6 hours a week.

# Functional Requirements

Must

* The system must save and load the student/faculty timetables.
* The system must save requests made by faculty members within the server.
* Students and faculty must be able to view a two-week timetable. This includes scheduled classes indicated by the module code.
* Admin must be able to create and alter timetables by adding, modifying or removing events; each event must have a:
  + Room location
  + Event name
  + Time & Date scheduled
  + Personnel allocation; faculty allocation is public, but student allocation must remain private.
* The system must store accounts persistently which require a username and password for all users.

Should:

* The system should allow students and faculty to receive additional information about a scheduled event. When selecting the desired event, the additional information provided will be:
  + Time & Date
  + Faculty name
  + Faculty email
  + Room location
* The system should allow faculty to request new events for the admin to review. Each request must include:
  + Request name
  + Room
  + Time & Date
  + Reason for request
* The system should allow faculty to request event modifications for the admin to review. This encompasses the removal of an event where each request must include:
  + Request name
  + Module code\*
  + Event ID\*
  + Modification’s to be made
  + Reason for request
* The system should allow faculty to request student reallocations to different groups within a module. When this request is made it must include:
  + Student ID number
  + Module code and group: current and prospective
* The system should display all requests made by a faculty member within a seven-day period on the client.
* The system should allow the admin to approve or deny requests sent by faculty members. Requests to be considered are:
  + Existing event modifications
  + New events
  + Student reallocations
  + Cancellations; which must allow the cancellation of timetabled events

If a request is denied the admin must state a reason and provide applicable alternatives\*.

* The system should display to the admin rooms that have the specified type and capacity for an event.
* The system should not allow students and faculty to have multiple scheduled sessions at the same time.

### Could:

* The system could automate requests which bypass admin review.
* The system could filter modify, remove and load requests to the automated approval sub-system.
* The system could generate logs by wrapping requests with a:
  + Time & date stamp
  + Unique ID
  + Status (default: Pending)
* The system could allow students to select events that they cannot attend and show the other available events for the same module. It will not display events that are full or that would clash with current scheduled events. If the suggested event is selected the system should update the predicted capacities of the effected events.
* The system could allow students and faculty to view their timetable offline.
* The system could schedule each event through a capacity optimisation algorithm to minimise the effects of over or under populating rooms that takes no longer than 5 minutes for all events.

## Non- functional requirements

### Must:

* The system must schedule all sessions between 9am to 5pm. However, special requests can be scheduled to extend to 7pm.

## Should:

* The system should be available for use 24 hours a day
* The system should support 120 simultaneous users which will be demonstrated using 8 students, 1 faculty member and the admin as confirmed by the stakeholders.
* The system should alert student and faculty of the changes to their timetables within 1 minute, or when they are next online.
* The system should notify the admin of requests within 1 minute.
* The system should run on Window, Linux and Mac.
* The system should use separate interfaces for student, faculty and admin to accommodate their privileges in the system.
* The system should use a single sign on approach.
* The system should use a consistent colour scheme.

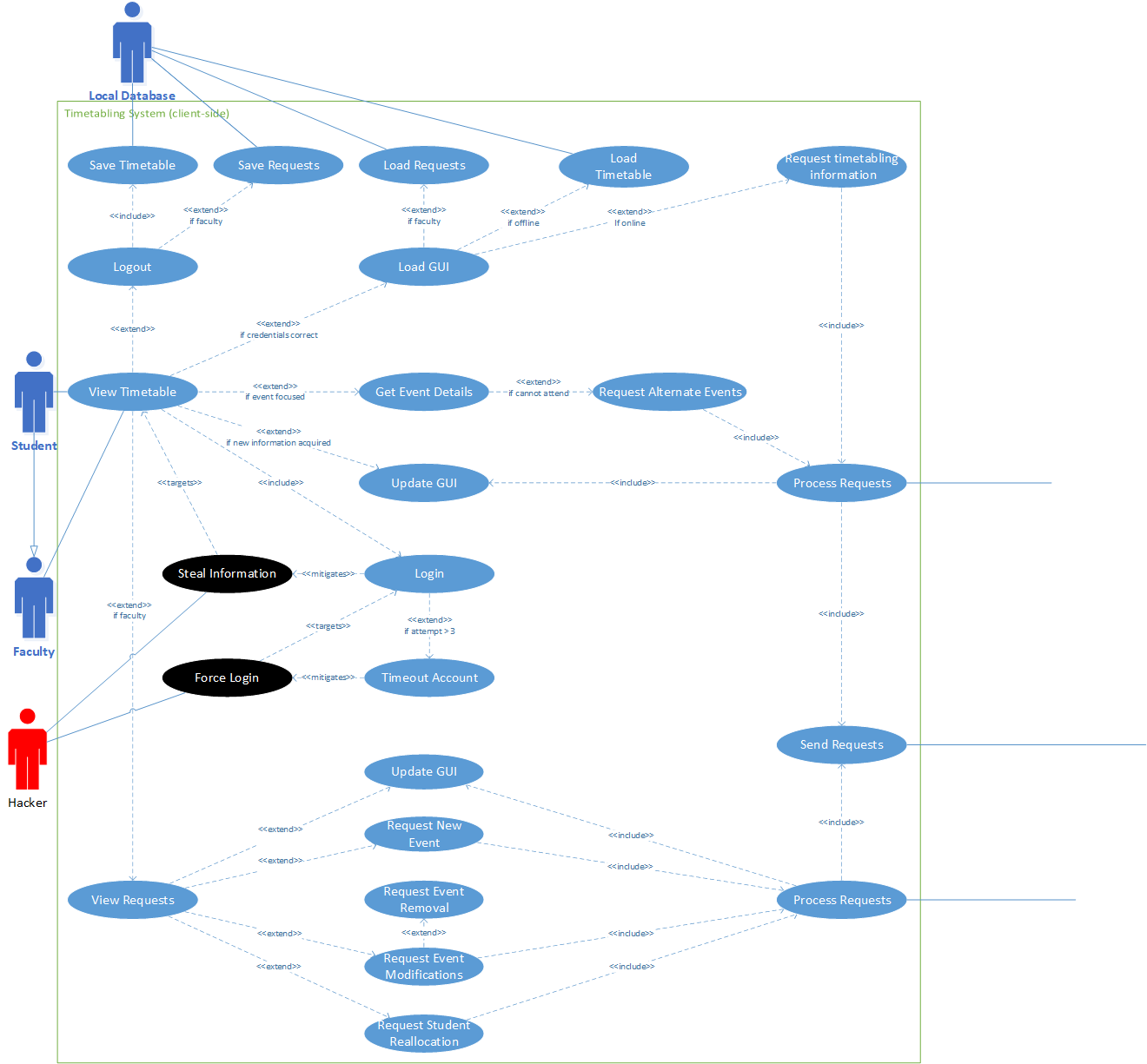
Could:

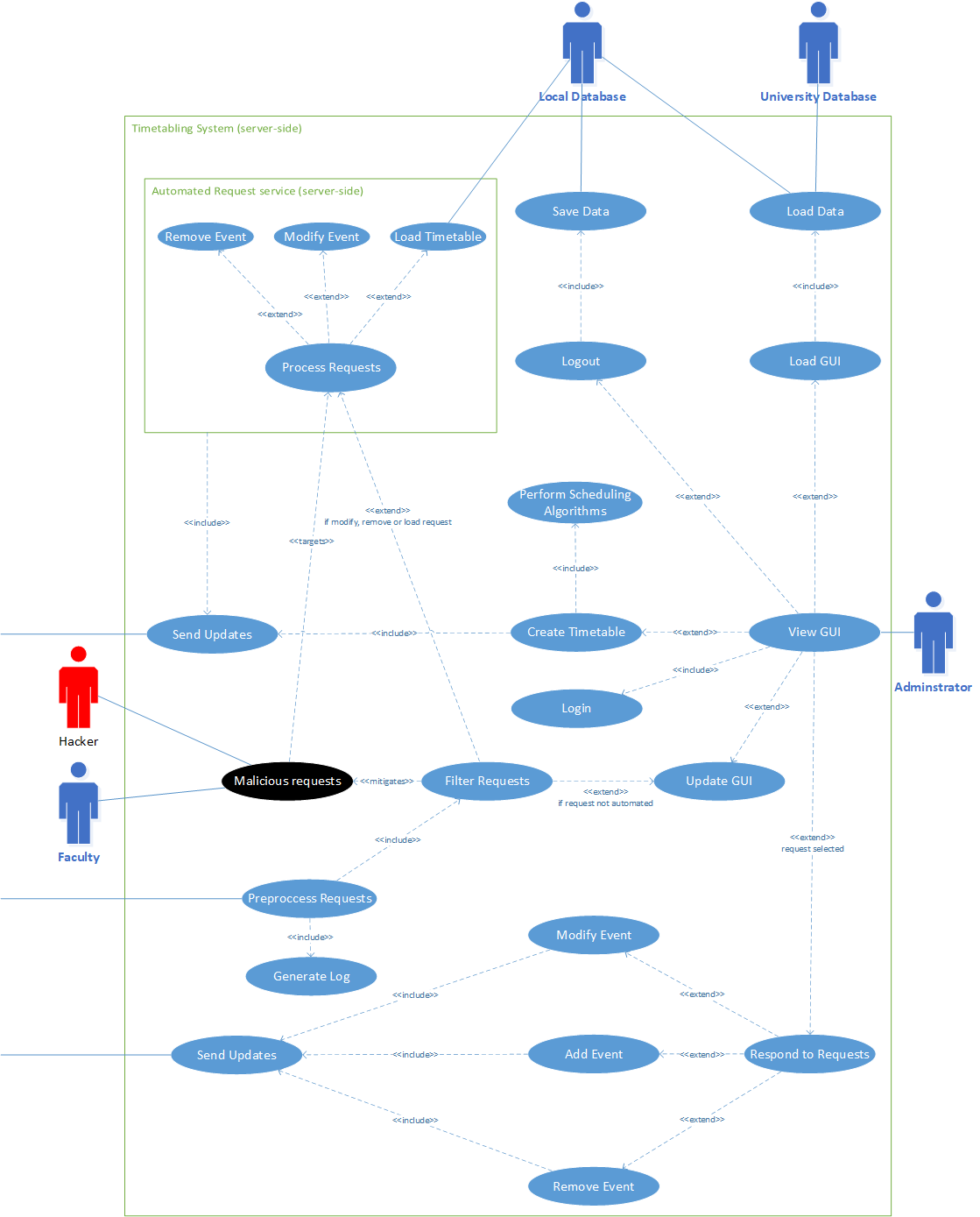
* The system could filter profanity.
* The system could reset timed out accounts at 12AM GMT +0.
* The system could incorporate a colour-blind mode.
* The system could encrypt messages sent over the network.
* The system could encrypt database calls.

Use case/ mis-use case

* Hackers stealing information.
* Hackers brute forcing the login.
* Faculty/Hackers making malicious requests.

Use Case Diagram





# References

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 43-59.

ECPI University. (2018). *ECPI BLOG*. Retrieved from Why is Web Design Important : https://www.ecpi.edu/blog/why-is-web-design-important

KSTEC. (2006). *ILOG Scheduler*. Retrieved from KSTEC smart solution: http://www.kstec.co.kr/products/scheduler/default.htm

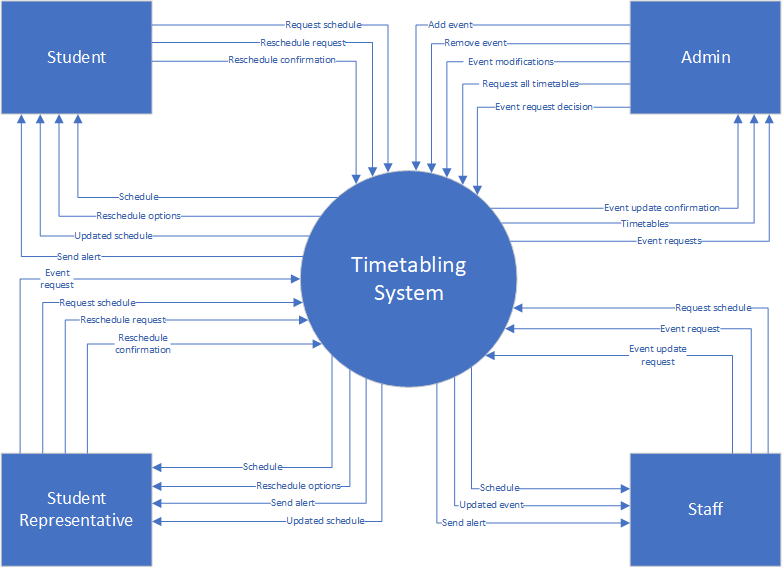
Nottingham Trent University. (2018, January 1). *Timetable - NTU Online Workspace*. Retrieved from NTU Online Workspace: https://now.ntu.ac.uk/d2l/lp/navbars/1/customlinks/external/15376

Pearson. (2015, May 5). PowerSchool SMS 8.0 eDocs.

Welling, L., & Thomson, L. (2003). *PHP and MySQL Web Development.* Sams Publishing.

# Appendices

## Appendix A: Context Diagram of System



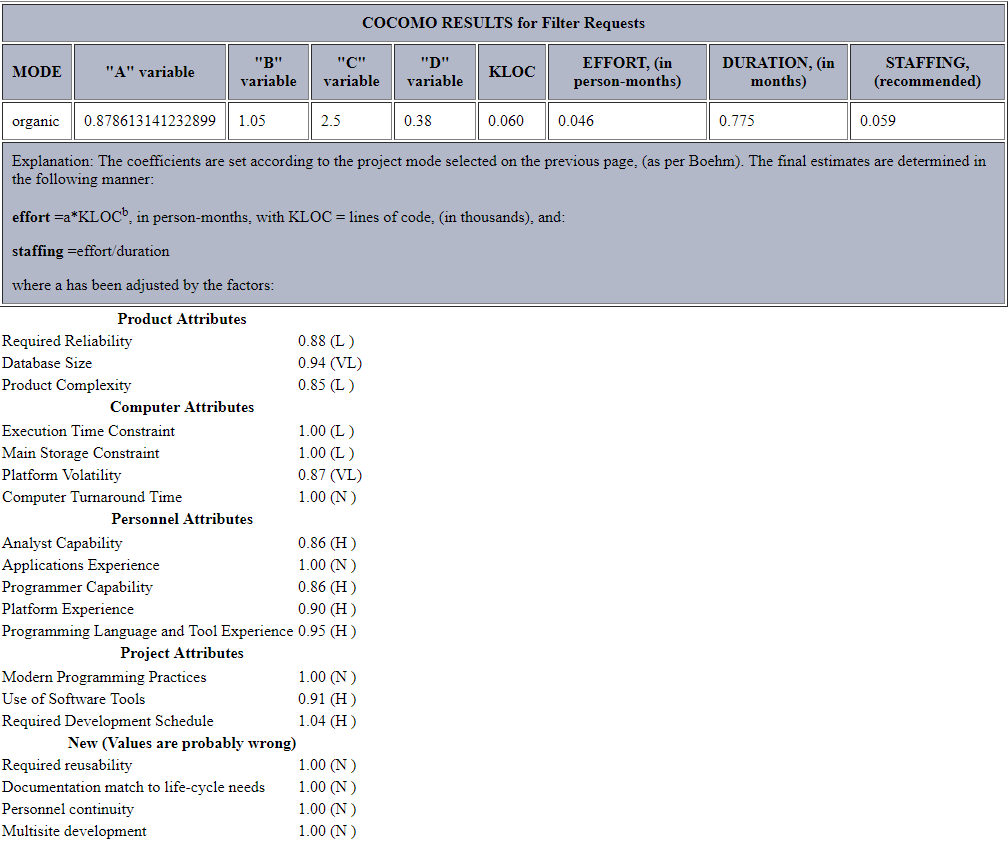
## Appendix B: Current Solutions

The existing solutions are listed below:

* ILOG Scheduler: a software that provides a solution through specialized models and algorithms for the scheduling of tasks and resources over a period. It includes a range of functions e.g. capacity scheduling, flexibility and extensible scheduling as well as fast and efficient algorithms and heuristics. It also has the capability of computing schedules that satisfy one’s objectives and constraints. (KSTEC, 2006). The ILOG scheduler is tailored to businesses to use during the planning stage of their projects, rather than for students and faculty. Though most of its functionalities are required to be incorporated in the timetabling system proposed by the stakeholder.
* Vcita: A scheduling application for small businesses. The application allows for synchronisation of the users work schedule with their calendar on any device. The capability of managing and viewing the team's calendar is also given to the users. The user can also allocate faculty to services and appointments which they can view in their personal calendars. As expected, it takes into consideration constraints such as customer cancellations, scheduling and rescheduling of appointments. Like the ILOG scheduler, Vcita is for the use of small businesses rather than faculty and students in academia. The integration of the various platforms the application could be used is a novelty for the users, as faculty can view the schedule wherever they are, and clients can book appointments anywhere.
* I360: an analytical software which works hand in hand with a Work Force Management (WFM) system. The core functionality is the ability to forecast a day’s workload for companies within the field in which they receive inbound contacts, i.e. customer service roles which are being contacted by customers. The system factors the colleague’s availability, their skill set, hours available and constraints. Throughout the day it is possible for the timetable to be manually adjusted and for tasks/breaks to be changed or moved. The system will proceed to analyse everyone logged into the system by monitoring their current activities to provide analytical data such as the current response rate to the incoming contacts in relation to the expected forecast for the day. The flexibility of handling constraints and quickly adjusting the schedule in response provided by the i360 is required to be included in timetabling system.
* PowerSchool: a system providing the service of scheduling for schools, by giving the school administrator the capability of building a master schedule. Within the master schedules the school administrator can experiment with various scheduling scenarios thus analysing and adjusting student requests, teacher allocations etc. The interface gives the opportunity for school administrators to schedule by course, teacher or room whilst also allowing for considerations of constraints by students or faculty. (Pearson, 2015). The core functionalities of the PowerSchool are like what has been requested by the stakeholder for the timetabling system. PowerSchool is seemingly tailored to American students in their secondary stage, contrasting with the timetabling system for students in higher education in the UK. Visually the interface looks simple but outdated regarding the design. The aim is to design the timetabling system to be aesthetically pleasing with ease of use for the user.

## Appendix C: Effort Estimation

The effort estimated using COCOMO for the use case of Filtering request is calculated as follows:



The calculations state that it would take 0.775 person months to complete the use case of filtering requests.