

Project descriptions

A summary of development projects, both in college and private

Knut Lucas Andersen

January 29, 2017

Contents

1	Introduction	3
2	Visual Studio: C#	4
2.1	Management of Databases	4
2.1.1	Oracle	4
2.1.2	C#	5
2.2	Bachelor Thesis: CleanMyFolder	5
2.3	Computer Graphics	6
2.3.1	Graded Assignment: Car game	7
2.4	Model Train registration	7
2.5	Multipurpose editor	8
3	Visual Studio: C++	9
3.1	C++ course	9
3.1.1	Graded assignment: Recipe Program	10
3.2	Machine Learning library	11
3.3	Computational Forensics	11
4	Java and Android	13
4.1	Animal Registration	13
4.2	Android course	13
4.2.1	TracknHide	15
4.3	TrackMyTeacher	16
4.4	BeregnSnitt	16
4.5	Auto-reply for Android	17
5	Python	18
5.1	Global Software Development	18
5.2	Advanced Project Work	19
5.3	Master thesis: Predicting coding question quality	19
6	Web development	21
6.1	Alumni Community	21
6.2	Flea Market	22
6.3	Applied Computer Science Project	23
6.4	Integration project	24
7	Others	25
7.1	Teacher-Student questionnaire system	25
7.2	Hotel management system	26
7.3	Flash Programming	26
7.3.1	Space Invaders	26
7.3.2	Tower of Knowlegde	27
7.4	3D Modelling	27
	Index	29

1 Introduction

The following document contains a summary of all the development projects I have been a part of, starting from my Bachelors degree until my completed Masters degree in Applied Computer Science. Each section is dedicated to a "main" programming language, but more than one may have been used in the development.

To ensure that projects covering more than one IDE, Editor, Tools or Programming language(s) is covered, the last page contains an Index listing these. Projects uploaded to GitHub will also contain a link to its given repository, but group projects are not available on my GitHub account.

GitHub Repository: <https://github.com/klAndersen?tab=repositories>

Some projects have been pre-described tasks/assignments from the professor of that given course, so some descriptions may seem a bit "artificial". This is because I have re-used the original assignment descriptions in some sections to make it more descriptive. Projects developed within a specific course will include 'College/University', 'Level' (Bachelor or Master) and the belonging 'Course'.

Revision history

As this document is a work in progress, you can find the latest version here: <https://github.com/klAndersen/ProjectDescriptionsRepo/blob/master/src/Project%20Descriptions.pdf>

Version #	Description of change (why, what where - a few sentences)
1.0	First version (temporary) finalized; need to write more detailed on my work in Section 5.1

2 Visual Studio: C#

2.1 INF313 Management of Databases

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF313 Management of Databases
- IDE, Editor, Tools: Visual Studio, Oracle SQL Developer, db4o¹, Prevayler (Bamboo.Prevalence)².
- Programming language(s): Oracle (first Assignment), Visual Studio/.NET C# (second Assignment)
- Oracle-Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Databases/INF%20315%20-%20Oracle/Oblig1>
- C#-Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/Oblig%20Inf315>

In this course, we were given two mandatory assignments that were graded. Both assignments was based on solving the same task, but using different development tools. The assignment contained an UML-model to describe the expected architecture.

2.1.1 Oracle

A real estate agent wants an Oracle database for registering sales objects that contains sellers, buyers and bids. For each buyer/property, only one bid is stored, and the buyer is bound by the bid until the deadline for acceptance expires. Therefore, bids cannot be withdrawn. If the buyer wants to register a new bid (where the buyer already has an existing bid), the bid would be accepted only if:

- the acceptance deadline on the previous bid expired
- the new bid is higher (greater amount) regardless of the acceptance deadline³
- the new bid has the same amount, but later acceptance date

It is possible to sign up as stakeholder without submitting bids, but everyone who bid will automatically be registered as well as stakeholder (if they are not already there).

For a given buyer, it should be possible to get all current bids, ie. bids where the deadline has not yet expired. When checking for bids that expires the same

¹ Open source object database for Java and .NET developers.

² <http://bbooprevalence.sourceforge.net/>

³ The acceptance deadline specified only by date - bid comes out throughout the day (until 24:00).

day, these should be returned with a the string "Urgent". In all other cases, it should return "Not urgent" - even if the time limit has expired. It should be possible to return all current bids (deadline has not expired) from a given buyer, which lists the property information. There is always only one vendor for each property. If the property is co-owned, the registered name should contain the person representing the owner. When calling the `to_string()` function, all information stored in the given object should be returned.

2.1.2 Visual Studio/.NET C#

A real estate agent needs an object-oriented application for registration of properties and bids on these properties. An UML-model was given representing the architecture, along with the client side of the application (using Windows Forms as GUI). An assumption we could make was that all objects would fit in RAM simultaneously, and we were tasked to create two different version. Both versions should ensure persistence by using two different libraries, the first using db4o, the other Prevayler (Bamboo.Prevalence).

The architecture for the system was a three-layer model. Communication went from the Menu (Windows Form; GUI) to a Controller class⁴. In the Controller class, data was checked/verified before being saved to file. If there were any errors, error messages were to be sent back to the user.

The specific entity classes was persistent, but not the control - and limit classes. Attributes marked with ID by users were to be identified for unique for each object that were stored.

There were also two notable differences that had to be taken into account with db4o and Prevayler. When persistence was secured with db4o, the SortedList containing the properties was unnecessary and therefore not included. When using search (find-methods), the search had to be made directly in the object database. With Prevayler the search had to happen in the real estate list since it does not have an object database.

2.2 Bachelor Thesis: CleanMyFolder

- Level: Bachelor
- College/University: Buskerud University College
- Course: INF350 Bachelor Thesis
- IDE, Editor, Tools: Visual Studio/.NET C# 2010, UMLet⁵ (to create UML diagram), PSPad (creating HTML help files), Microsoft HTML Help Workshop (for compiling help files), SourceForge, CygWin
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/Bachelor%20thesis>

For selecting a Bachelor thesis, we had two options. The first was to do research and conduct a scientific study. The second was to develop a system based on a

⁴ The Controller class mainly handled adding, deleting, updating, retrieving and searching for data provided by the user.

⁵<http://www.umlet.com/>

given problem/situation (the option I selected). My thesis was based on the following problem: "I want more space on the harddisk - which files do I delete?"

The delivered system was a file management system which could scan the computer for files and folders at a given location, and then list them in separate views. It was also planned to include the possibility to scan for identical duplicates and check for duplicates based in zip-files, but this was not included due to time-restraints.

A user may not be interested in small files, or perhaps files that were created as early as last week. Therefore the menu for initializing a scan includes several filtering options (include subfolders, file type, size and date). When a scan was completed, the user would be presented with the results, where the folders were listed in a TreeView and files in a DataGridView (presuming there were any results). The structure in the TreeView listed the parent folder (path selected) as the root, where every subfolder (and their subfolders) were listed as a child of that parent folder. When selecting a given folder, all the files in that folder was listed in the DataGridView (file name and - type, size, date created/modified/accessed).

In addition to getting a listing of folder(s) and file(s), the user could open, move or delete the selected folder/file(s). For deletion, there were two options; move to recycle bin, or delete permanently. However, if a filtered scan was ran, the folder deletion options was disabled. This was done to avoid the deletion of a non-empty folder (e.g. only textfiles were listed, but it also contained PDF-documents).

2.3 ITE1605 Computer Graphics

- Level: Bachelor
- College/University: Narvik University College
- Programming language(s): C#/XNA
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/XNA/Datamaskingrafikk>

The goal of this course was to get an understanding on how computer graphics worked both in simulations and game development. There were a total of five programming assignments, where the last one was part of the course grade. The first two assignments were based on understanding the use of the XYZ-system, by using matrices and lines to draw a 3D-Cube and a spinning 3D-ball. The fourth assignment included use of particle effects and lighting/shaders.

In the third assignment, the task was to create a simulation of our solar system (this I did not manage to complete properly). The sun should be in the middle (and be the largest object), with the planets rotating around the sun (in addition to its own axis). The planets should also have moons with them, but the sizes and distances could be approximations (but one should attempt to make this as realistic as possible).

2.3.1 Graded Assignment: Car game

The graded assignment was a topic of our own choosing, but it required that we implemented at least one element of all the topics taught in this course. I chose to develop a car game, where I used the 3D-models I had developed in the 3D-modelling course (described in Section 7.4). This was a very simplistic game, where all the player could do was drive around in a "stadium" to gather flags (which gave the player points; static position). The flags gave the player points, but the score was not stored, and was reset when starting a new game. The player could not drive out of the stadium, nor could the player destroy or crash the car.

The game included music and sounds (e.g. screeching sound when breaking), which could be turned on/off (three different background musics were available). The camera followed the car, but the player could swap with camera being behind the car or in front (front-view look). The particle effects were present as exhaust from the car, and by using gears, the player could adjust the speed (1-4, and reverse).

Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/XNA/Datamaskingrafikk/Karaktergivende%20oppgave>

2.4 Model Train registration

Project type: Private

Programming language(s): Java⁶, Visual Studio/.NET C#.

One of my friends interests was model trains. As he started to get a larger collection, he wanted to get a better overview of what he owned, and also the possibility to create a simple wishlist to give to friends and family.

He wanted a separation between whether the model he registered was a train, a wagon or an accessory. Trains and wagons had nearly the same specifications, but accessories had a wider group, as these could be railroads, trees, figurines, houses, etc. Since he could own more than one wagon, instead of re-registering an existing wagon, he wanted to just update the amount of wagons owned. When creating a wishlist, he wanted to be able to save it to a textfile for printing. The wishlist should contain the models number/ID, name, sales location and price.

The finalised version ensured persistence by using Prevayler (Bamboo.Prevalence), which takes snapshots of the current state of the program, its objects and data. With Prevayler, he had no need to install any excess programs, and if there was a lot of useless data stored, it could easily be removed by deleting the existing file(s). In addition to desired requirements, the program also included the ability to search for models (based on number/ID, name or type). To ensure that models could be easily updated when listing registered models, these were presented in an editable DataGridView.

Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/ModellTog>

⁶ The project was originally developed using Java, but he was unable to make it run. It was therefore re-written using C#.

2.5 Multipurpose editor

- Project type: Private
- IDE, Editor, Tools: Visual Studio/.NET C#, ODBC (for MySQL)
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%23/MultiPurpose>

At the time (2011-12), I often experienced MySQL Workbench being slow at start-up, and sometimes I wanted to just a quick peek at the tables or data in the database. Creating Readme's for programs and systems could also be slightly time-consuming, so I created this simple program I named "MultiPurpose". This program had three functionalities:

1. Text Generator: When creating experience level requirements for games, setting these values can be quite time-consuming. The same can be said if you are testing a lot of "random" links with a set incremental numeric value. Here one can enter a start/end text, and append numbers (min, max, step). If you are testing links, you can select which browser to open the link in (converted to links if the text contained 'http:/').
2. ReadMe Generator: Saved the text you entered to a text file. Options include overwriting existing file, adding asterics, and adding version number into the text.
3. MySQL Display: Establish a connection to a MySQL user account. After entering valid username, password and host/port, you could display the information available for the given users database(s). You could select a database, select a table, and run some queries against the selected database.

3 Visual Studio: C++

3.1 ITE1546 Programming C++

- Level: Bachelor
- Project type: Assignments
- College/University: Narvik University College
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%2B%2B/ITE1546%20Programmering%20i%20C%2B%2B>

This was an introductory course, targeted at the Electrical engineering students. Therefore it was only commandline and no GUI, using using native C++. There were three mandatory assignments, and one graded.

Assignment 1

The first assignment was focused on understanding the basics, and creating a simple commandline menu:

1. Calculate the volume of a ball/sphere (based on entered radius).
2. Solve a quadratic equation (based on entered a, b and c).
3. Calculate distance between two geographical points (using Haversine Formula).
4. Calculate and print repayment plan for annuity loan. The user enters the home loans, interest rates (in %) and the number of years for repayment.
5. Convert a number (integer) to text (e.g. 31 = thirty).

Assignment 2

The second was based on creating a simple cinema ticket system, which only had to register one performance (the user was the ticket seller). The cinema contained 15 rows, and 30 seats, where vacant seats were marked with '*' and sold seats were marked with a '#'. The cinema operated with two prices: full price (e.g. £9) and reduced prices (e.g. £6). The menu with its requirements are listed below:

1. New performance: Sales data was stored in textfile, so that it could be re-used on startup. This file was only reset when selecting this option. This option set all seats to vacant, and asked for the name of the new performance.
2. Buy Tickets: Allows user to select desired row and column for seating, in addition to registering which price the ticket(s) were sold for. After confirming sale, the selected seats were marked as sold.
3. Show vacant seats: Prints a view of the current performance, showing which seats are vacant.

4. Display ticket sales for this performance: Show the number of tickets sold per day, income per row and total income on current performance.
5. End program

Assignment 3

The third was based on Bitmap pixel colour manipulation, where one should be able to alter the colour of all pixels, or grey-scaling them all. The assignment also required that they were based on the following two prototypes:

```
// component: 0 = R, 1 = G, 2 = B
void changeColor(vector<Pixel> &pixelTable, int component, int colour)

void greyScale(vector<Pixel> &pixelTable)
```

3.1.1 Graded assignment: Recipe Program

A chef wants a program to keep track of nutritional information related to meals that he composes. By using this program, he should be able to compose dishes and get calculated corresponding nutrition information as energy in kcal (and/or Joule) and the content of proteins, fat, carbohydrates, vitamins, etc. This data was extracted from an Excel file (converted into CSV-format), which contained nutritional values for more than 1300 different food types (per 100 grams of edible product).

The chef envisions that he chooses x grams of ingredient⁷, y grams of ingredient₂, z grams of ingredient₃, etc. to compose a dish and then (or continuously) get printed total nutrient for the new dish. The recipe contains quantity (in grams) for each food and overall nutrition information (amount of protein). Each new finished dish should be written to a recipe file (a separate file per dish).

The program should also print out different reports:

- All registered dishes/recipes listed on the screen.
- Dishes with more than x (determined by user) g protein, fat or carbohydrates per 100g done right.
- Dishes with more than x (determined by user) calories per 100g finished right.
- Dishes with lots of vitamin D (compared to the daily recommended intake).
- Dishes with lots of vitamin A (in relation to the daily recommended intake).
- Dishes with lots of cholesterol (compared to the daily recommended intake).

Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/C%2B%2B/ITE1546%20Programming%20i%20C%2B%2B/Karaktergivende>

⁷ The given ingredient is selected based on its food number.

3.2 Machine Learning library

- Level: Master
- Project type: Developed for re-usability in assignments
- College/University: Gjøvik University College
- Course: IMT4612 Machine Learning and Pattern Recognition 1
- Repository: <https://github.com/klAndersen/Machine-Learning/tree/master/MachineLearning>

This was written for use in the course "IMT4612 Machine Learning and Pattern Recognition 1". It is not a complete project, as it is mostly built on the algorithms taught in the course, and those that were asked for during the mandatory assignments. Aside from algorithms, it contains a Conversion class (string to numeric, native to Managed C++, vica versa, etc), a Matrix class, a class for Statistics and Exceptions.

3.3 IMT4641 Computational Forensics

- Level: Master
- Project type: Assignment
- College/University: Gjøvik University College
- IDE, Editor, Tools: Visual Studio, SQLite, Microsoft HTML Help Workshop
- Repository: <https://github.com/klAndersen/IMT4641-Computational-Forensics>

This course was development only, where we were tasked with coming up with our own idea to develop a tool or system that was relevant to the courses we had. My project focused on creating a re-active tool (used after an incident has occurred) that analysed Android databases (SQLite). A requirement was that the database(s) was already extracted from the given device(s) that were under investigation.

After selecting a location containing database(s) to investigate, the user has the option to add filters to their search. Since the path could contain more than one database, they were given the option to select all or just a few. Search categories and types could also be defined (e.g. a category could be "Internet", and a type could be ".html"). In addition, option to select a timeline (start/end date) were also included. This way, if there was a specific day or week being investigated, this could reduce the amount of results that needed analyzing.

The result screen contains a TreeView which lists the folder(s), database(s) and the database(s) table(s). The folder is the parent node, where the database is a child of the folder and the table(s) are the child(ren) of the database. If the investigator was looking for something specific, the investigator could search through the TreeView. Matches were then highlighted in yellow.

The content of a given databases table was listed in a DataGridView, which listed the columns and rows. Databases that were empty, only contained one row in the table 'android metadata', or did not match the search criteria(s) was excluded from the result. If the investigators found any databases they wanted to take a closer look at, they could just rightclick the database in the TreeView, and select "Open folder".

4 Java and Android

4.1 Animal Registration

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF240 and INF244 Object Oriented Programming
- IDE, Editor, Tools: Eclipse, MySQL Workbench
- Programming language(s): Java
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/Oblig2_Inf240_KnutLucasAndersen
- Repository: https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/Oblig_Inf244_KnutLucasAndersen

In the two Java courses in my Bachelor, we had the same assignment, but with different requirements. The main focus was to register two different animals (hares and lynx). Common for both was the registration of their gender (male/female), length (Double), weight (Double), time of capture (day, month, year) and location (String). The ID was to be incremented, and start with the initial for the given animal (e.g. L1, L2, . . . , H1, H2, . . .). For the hares, colour (String) and type (Char) was to be registered. For the lynx, the length of their ears (Double).

If an animal was later re-captured, their data should be updated instead of re-registered. The program had the possibility to search for animals based on ID, show amounts of re-captures (based on the entered year), show amount of different hare captures (based on type), and an unsorted report of all first-time captures.

The first course was introductory level, so input/output was shown in command line, and data was stored in a text file. In the second course, the user was presented with a GUI that we had to code ourselves (not use a Designer), and the data was stored in a MySQL Database (using JDBC as database driver). Singleton was used to ensure that there was only one object maintaining the database connection. The main menu used JFrame, and the child windows used JDialog.

4.2 ITE1621 Applications for mobile and web

- Level: Bachelor
- Project type: Assignment
- College/University: Narvik University College
- IDE, Editor, Tools: Eclipse, MySQL Workbench
- Programming language(s): Java, Java Servlets, Android 2.3 - 4.1, XML, SQLite

- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Android/ITE1621%20Applikasjoner%20for%20mobil%20og%20web>

This was an introductory course to developing applications for Android.

Assignment 1

The task was to develop a conversion app for different measure systems (distance, volume, mass, temperature and time). When starting the application, the user begins by selecting the measurement category, unit to be converted and then enters the numerical value. A requirement included the use of Fragments, since the application should be able to run on devices with small and large screens, and also multilingual (Norwegian and English).

Assignment 2

The task was to develop two different applications (application A and application B). Application A could be used to manage information about books/book titles, which was stored in a text file. Application A should at least contain:

- A Start-up activity functioning as a menu to access other activities.
- An Activity that displays a list of all registered titles. The user can then choose a book title and this should then be returned.
- An Activity that registers a new book title, and stores this in the file.
- An Activity that allows deletion of selected book title.

The activities created added in application A were then to be used by application B (e.g. title selection (which is returned from A to B), adding new titles, etc.). For each new title added, a broadcast is sent from application A, which is to be caught by application B and displayed in a Toast.

Assignment 3

The task was to develop two Android applications, one that logs and stores information about various "events" and one that serves as an interface to the log.

Application 1: Event Logs

This application could (by using a BroadcastReceiver) capture and store information from system events (e.g. receiving SMS, phone calls, GPS turned on/off, WiFi turned on/off, etc.). Each event had its own category (e.g. Telephony, Network, Positioning, etc.), and stored it in a SQLite database (which was "exposable" via Content Provider). The stored data for all events are date, time, category, text, details (parameters/values associated with the event, e.g. who the SMS came from). When the application is started, the user could see a "Setting" activity that allowed the user to decide what events to log. When a broadcast occurred, the BroadcastReceiver would start and store the relevant information in the database (the application did not have to be started for this to work).

Application 2: LoggerUI

This application displayed the information that was stored in the database by Application 1 (by using ContentResolver). The user could then select a category and display all events for the given category.

Assignment 4

The task was to develop a temperature monitor and notification application for Android, where data was downloaded from a weather website as XML. The data was downloaded by using a Service (which also handled the XML-parsing), that started when the device started. The service could also be started/stopped through the applications ActionBar, but unless stopped, the service continued running after the application was exited. The temperature data was measured by various weather stations, where each weather station have a unique number and name (e.g. '84700', 'Narvik airport'), but they can have the same URL. In addition to extracting the temperature, the weather station data was also needed so that users could select which weather stations they wanted to monitor.

In the application, the user could select between getting a list showing monitored weather stations, adding new weather stations for monitoring, and displaying application settings. For the list of monitored stations, the weather station number, name and last monitored temperature was displayed (and also had the possibility to remove selected station(s)). In the Settings, users could set download interval (e.g. every hour, every two hours, every three hours, etc.) and temperature range (upper/lower limit). After temperature data was downloaded and extracted, if the temperature for any given station was above/below set limit, a notification was given to the user.

4.2.1 TracknHide

This project was part of the course grade, and consisted of developing an application that could show the different users location and route on a map. The project had two parts, an Android application and a Java web server.

The Android application used Google Cloud Messaging (GCM) to send and receive data, and Google Maps (API v1) to present a map. By using MapOverlay, users could also show the adress when clicking a location on the map, and switch between Street – and SateliteView. On the map, the route and current position of logged in users with shared position was shown. Users could also choose to store their own or others route, which could later then be re-drawn on the map. The position data was retrieved by using the devices GPS, but the amount of position data sent was set by the user (amount of time passed or distance moved).

The Java Web server was created using HTML, Java Servlets and JSP, running on Apache TomCat v7.0. It stored the user related information in a MySQL Database, and kept track over all the currently logged in users. The web pages was mainly targeted at an administrator, to give them the ability to view connected users, and disconnect if needed. The server was also the connection point for the users to share/receive position data and notifications via GCM.

Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Android/ITE1621%20Applikasjoner%20for%20mobil%20og%20web/Karaktergivende%20oppgave>

4.3 TrackMyTeacher

- Level: Master
- Project type: Assignment
- College/University: Norwegian University of Science and Technology (NTNU)
- Course: IMT5401 Advanced Course in Mobile Technology
- IDE, Editor, Tools: Android Studio, MySQL Workbench
- Programming language(s): Java, Android, XML, PHP, Git, JSON
- Repository: <https://github.com/klAndersen/IMT5401-Mobile-Research>

In this course I tried to develop an indoor tracking prototype app for Android. The purpose was to be able to locate teachers and professors on campus, by measuring the WiFi-signals at their current location (if they had enabled/allowed tracking of their current location).

The server side of the application was developed using PHP (because the college server only ran PHP), which mainly handled data transfer between the Android app and the MySQL database. When data from the database was needed, the server retrieved the data, added it to an array as JSON and printed it as HTML. This HTML was then converted to a JSONArray on the mobile device (as I stated in my report, it would probably have been much easier if one could have used Java Servlets, since one could overwrite the HTTP GET/POST methods).

To get test data, I went to different buildings, rooms and floors spread out on campus (20 rooms, signals measured in all 4 corners plus the middle of the room). What I found when looking at the gathered data was that the measurement for the WiFi-signal was set too low (value=5). Therefore this prototype was not finished, as I did not have the time to re-measure all the rooms to get more accurate signal data.

4.4 BeregnSnitt

- Project type: Private
- IDE, Editor, Tools: Eclipse
- Programming language(s): Java
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Java/beregnSnitt>

When nearing the end of my Bachelor degree, I wanted to know what my average grade was, to see what colleges and universities I could apply to in regards to taking a Master degree. This led to the development of "BeregnSnitt". BeregnSnitt calculates the average score based on the grades and total points the user has.

The user is presented with a GUI (Jframe and Jdialog), where s/he can enter the amount of A to E's achieved, the course points and the total points achieved (e.g. Bachelor = 180). I also added the possibility to add additional course grades, in case some courses varied (e.g. if the standard was 10, but one course had 3 points, another 5, etc.).

4.5 Auto-reply for Android

- Project type: Private
- IDE, Editor, Tools: Eclipse, Android Studio⁸
- Programming language(s): Java, Android, XML

This was a program I started on after learning Android. The reason I started working on it was because one often gets text messages or calls when you cannot reply or answer. Sometimes, people continue to text or call, and I therefore started working on what I call an Auto-reply app. The auto-reply is sent out as a text message (SMS), which could either be random or a set topic.

The auto-reply could be sent to only to those marked for it, or apply to all in the phone's contact list. The auto-reply was restricted by a delay, adjustable through the settings. When a call or SMS came in, if the person calling/texting was on the Auto-reply list, a timer would start. When the timer ran out, a check would occur to see if the SMS/lost call was unread. If so, a SMS would be sent out.

The SMS that were sent from the Auto-reply are random, but that was mostly because I created it for myself. I planned to later add some Machine Learning logic to it, so that it could reply somewhat intelligently to the SMS, instead of a random text.

⁸ The development started with using Eclipse, but was later changed to using Android Studio as IDE.

5 Python

5.1 IMT4112 Global Software Development

- Level: Master
- Project type: Group, Assignment
- College/University: Norwegian University of Science and Technology (NTNU)
- IDE, Editor, Tools: PyCharm, Arch Linux (OS), VirtualBox, OpenEdX⁹, XBlocks¹⁰, Git, GitHub, Scrum
- Programming language(s): Python, HTML, JavaScript

Project background

OpenEdX is a community driven, open, open-source development and delivery platform for online courses and MOOCs. The codebase is hosted on GitHub under the edx-platform repository, and managed by the EdX consortium. There is a native Android client and iOS app codebase hosted on GitHub too. The project offers comprehensive learning management system tools and authoring for lecturers, allows uploading the lecture videos, lecture slides, audio and preparing written course material in a form of handouts. There is a number of built-in modules for various in-class tasks.

Project goals

The goals of the project are three-fold:

1. to understand and gain in-depth knowledge about the platform, architecture, and organization of the existing codebase
2. to understand the process, requirements, and compliance when contributing to a (large) open source project
3. to be able to contribute to an open-source project

Task 1

Multi-choice questionnaire with confidence level attached to each individual questions that changes the marking of that question. We will follow a 3-levels-of-confidence model, in which the following is the scoring matrix for appropriate confidence levels (the "+1" means: "student gets 1 point for correctly answering this question"):

confidence	correct answer	wrong answer
0 - low	+1	0
1 - normal	+1.5	-0.5
2 - high	+2	-1

In this task, the parts I worked on was the creation of answer alternatives, the answers submitted by the students and and grade for the submitted questionnaires (with confidence level).

⁹<https://open.edx.org/>

¹⁰<https://open.edx.org/xblocks>

Task 2

Short-answer or multi choice questionnaire that allows students to annotate questions with two values: "for all students in this class, this question is harder than average" or "easier than average". This should allow the students to reflect on the knowledge of their peers and to judge the level of confidence of their peers. This point can be extended by calculating the scores for people that got more of the annotation correct as calculated against the actual test results for this particular question in this test.

Here I worked on the part related to difficulty level, and updating the students score.

5.2 IMT5251 Advanced Project Work

- Level: Master
- College/University: Norwegian University of Science and Technology (NTNU)
- IDE, Editor, Tools: PyCharm, MySQL Workbench, Arch Linux (OS), Git, GitHub, Py-StackExchange¹¹
- Programming language(s): Python 2.7, HTML, Django, XBlocks
- Repository: https://github.com/klAndersen/IMT5251_AdvProjWork

Preliminary work for my Master thesis, which was mostly focused on creating a simplistic prototype. The topic for my Master thesis was to develop a Chat Agent that could answer students programming questions, and also help them learn to be better at asking good questions. The Chat Agent was to be implemented as a module (an XBlock) in the Learning Management System (LMS) Open edx.

During this development, a simple chat interface was created which could connect to StackOverflow.com. One could ask questions, but it only returned the first question/answer it found on Stack Overflow. The prototype was also presented to two students that gave feedback and suggestions for features. The plan for further development was to implement AI in my Master thesis using a hybrid algorithm of Hidden-Markov-Model and Bayes Net.

5.3 Master thesis: Predicting coding question quality

- Level: Master
- College/University: Norwegian University of Science and Technology (NTNU)
- Course: IMT4904 Master Thesis
- IDE, Editor, Tools: PyCharm, MySQL Workbench, Arch Linux (OS), Git, GitHub
- Programming language(s): Python 2.7, Python 3.5 (submitted version)

¹¹<https://github.com/lucjon/Py-StackExchange>

- Code Repository: https://github.com/klAndersen/IMT4904_MasterThesis_Code
- Thesis Repository: https://github.com/klAndersen/IMT4904_MasterThesis_LaTeX/

During the presentation of the thesis topic, I was told that my topic was not just too large for a Master thesis, but even too large for a Ph.d. thesis. It therefore needed to be reduced drastically, which ended up with the topic being: "Predicting coding question quality using Stack Overflow ratings".

Problem description

Most of the systems that have been developed so far focuses on finding the best answer to a question asked by the user. Few, if any, focuses on the quality of the question being asked. What defines a good question, and can we in anyway predict whether or not a new question posted on Stack Overflow will be considered good or bad by the community? There are many users who has either a negative view or relationship in regards to Stack Overflow. Many experience that their questions gets down-voted, closed or even deleted. For some, they simply do not know how to ask an acceptable question. Questions related to homework are one example of questions that are not accepted on Stack Overflow. There is even a post on Meta.StackExchange discussing whether or not it should be acceptable to use greetings and sentiments in posts¹². Therefore, the question becomes: What is and is not a valid question on Stack Overflow?

Research questions

- What defines a good (coding) question on Stack Overflow?
- Can we predict a questions quality by using Support Vector Machines (SVM)?
- What type of features increases the accuracy of the SVM?

Development process

The data used in the thesis was downloaded from StackExchange Archive¹³, and inserted into a MySQL database, where the tables were based on the files available. Data was extracted from the database into Python using a library called pandas¹⁴, which also had the ability to export the data to a CSV file. The questions were originally in HTML format, so they were therefore cleansed using BeautifulSoup 4¹⁵ (bs4). To get the vocabulary over all the words used in all the selected questions, CountVectorizer from the Machine learning library scikit-learn¹⁶ was used. Features were extracted and replaced with feature detectors, and these were then used to create a model for predicting the question quality.

¹² <http://meta.stackexchange.com/questions/2950/should-hi-thanks-taglines-and-salutations-be-removed-from-posts>

¹³ <https://archive.org/details/stackexchange>

¹⁴ <http://pandas.pydata.org/>

¹⁵ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

¹⁶ <http://scikit-learn.org/stable/>

6 Web development

6.1 Alumni Community

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College
- Course: INF268 Internet Application Development
- IDE, Editor, Tools: PSPad, Apache XAMPP, MySQL Workbench
- Programming language(s): HTML, CSS 3, PHP, JavaScript
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Web%20Development/Inf%20268%20Utviklingsprosjekt>

This was a course that focused mainly on development for Web. In this course, everything was developed from scratch, without use of external frameworks or libraries. The goal was to develop an Alumni community web page for the college. A requirement was that the pages should look as identical as possible regardless of the web browser that was used (I compared Chrome, Internet Explorer, Opera and Firefox).

The course was split into two semesters, and the development project into five parts:

1. Web page design/layout and policy
2. Access Control
3. Functionality and database model
4. Posts and user administration
5. Completion

The users were divided into the following user groups; registered, moderator and administrator. Administrators was able to appoint new moderators and edit users, in addition to having the same rights as moderators.

All registered users should be able to...

- ...register, login, update their profile and be able to change/request a new password (if forgotten).
- ...upload a profile picture, or select one of the existing ones that had previously been uploaded to the alumni page.
- ...join or leave networks created within the alumni community. Users should be able to message each other, and also see a notification when new messages has arrived.
- ...see a list of the other registered users.

Moderators should be able to...

- ...edit users profile (e.g. if inappropriate data was entered on their profile).
- ...notify users they had breached the guidelines or block out/quarantine users (users should not be deleted).
- ...create/edit/delete existing topic fields (e.g. events, area of expertise, etc).

In my project I mostly used PHP and HTML. Every page was created as a .php, as this allowed me to update the page layout in just one file, instead of having to update all the pages. The layout was then included at the location where it was to be used. It also simplified the use of function calls and checks to see if a user was logged in, and had the proper rights to view a given page. Database access was also handled via php, and JavaScript was not largely used (mostly to verify user input).

The following is a short summary of what the delivered version included. For new users, they had a registration form, that gave feedback both with JavaScript and PHP if an error occurred. If the user continued, even though one or more fields contained errors, the given field(s) would have a text marked in red next to it, explaining what was wrong. Users also got their own profile, which allowed them to update their information, alongside changing their password (users that weren't logged in could request a new password if they forgot it). Moderators also had the ability to edit user profiles, e.g. if any added something in-appropriate.

Since users should be able to send messages to each other, they also needed to be notified if they had received a new message. This was accomplished by updating the link to the messages in the navigation bar with a number, where the number indicated how many new, unread messages they had received. Users could also message each other via their profile, or by searching for them from the member list (which listed all, or those matching the search criteria; name or e-mail). Users who broke the page policy could be put in quarantine. This rejected the users attempt to login, and giving them a message that they were quarantined. An information message could also be set, that would explain to the given user why they were quarantined.

Moderators could register campus, area of expertise and networks (e.g. degree, courses, etc). Moderators could also create events. All of the aforementioned could be updated or deleted. For events, only those active were displayed (but administrators and moderators could see all on the creation page). Events that were out of date was not showed, and this was also true for future events (based on start date for the given event).

6.2 Flea Market

- Level: Bachelor
- Project type: Assignment
- College/University: Buskerud University College

- Course: INF329 XML
- IDE, Editor, Tools: XMLEditor
- Programming language(s): XML, XSD, XSLT, PHP, CSS 3
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/Web%20Development/Inf%20329%20XML%20-%20blig.%20oppg>

A school band arranged yearly a flea market, to increase their income. This was a process lasting for months, starting with organization of the sales, to finally selling the various sales items. The flea market consists of several departements, several people are involved and also a lot of money. Income overview and control is a necessity to be able to plan next years flea market. Up until now, they used Excel sheets to keep an economical overview (participants sorted by departements). Every participant gets a bag of change, and can deliver income several times throughout the day. The participants on a Saturday is not necessarily the same as those on a Sunday (sales were only on Saturdays and Sundays). A daily overview over departement sales is registered, and a control is done to check the money delivered against the registered income.

The goal was to re-write this web-application to use XML as data storage, as the currently database in use (Access) gave compability issues depending on the version used. The assignment contained attachments showing screenshots of the Web-page, the Excel file and how the Access database looked. A requirement was that the file containing the database data should be in XML, and that this file should be controlled by using XSD (since the XML file could be altered manually). It was also a requirement to create at least four different web-pages to show how they could use the created XML-files to present and store new data.

6.3 IMT4003 Applied Computer Science Project

- Level: Master
- Project type: Group, Assignment
- College/University: Norwegian University of Science and Technology (NTNU)
- IDE, Editor, Tools: NetBeans, MySQL Workbench, Git, Scrum
- Programming language(s): Java Servlets

In this course, our group worked on developing an A/B Test system. The following is a short project description, taken from our group report:

"The product is an Internet application, where the goal is to track user interaction. The developer/administrator is able to see the user interaction real time, and the interaction will also be stored in a database, to be shown for the developer/administrator later. This functionality is relevant when a person or a company wants to know the best possible user interface for a website. With this Internet application, you can present an interface for a user group, and then

you can compare more interfaces with each other, to decide which one the user interacts with most efficient.”

Various frameworks were used in this project, but I those I was directly involved with was the Java Servlets and MySQL. In the development part, I had the responsibility for the database model, the database and the database operations. Since we used Model-View-Controller (MVC), some of the programming I did was related to the interfaces and connections between the GUI, the Controller and the Database.

6.4 IMT4004 Integration Project

- Level: Master
- Project type: Assignment
- College/University: Norwegian University of Science and Technology (NTNU)
- IDE, Editor, Tools: Unity, Notepad++, MySQL Workbench, d3.js¹⁷, Git,
- Programming language(s): HTML5, CSS 3, PHP, JavaScript
- Repository: <https://github.com/klAndersen/IMT4004-Integration-Project>

The original plan for this project was to create a 3D-simulation game that incorporated Darwins theory of evolution (using Unity). The goal was to try helping students taking Machine Learning courses to learn an algorithm called ”Genetic Algorithms”. However, due to time restraints and complexity, this was changed and reduced to an unfinished web-page version (aimed at teaching Genetic Programming).

This version consisted of pages for students and teachers, where teachers could create new assignments, check student participation and adjust student scores based on their answers/progress. The students had access to creating and editing their profile, see game progress and score, and they could play a given game an unlimited number of times (but attempts and answers were logged in the database). After submission, the given game would no longer be playable.

The only game-element that was implemented was a LISP (reverse polish notation) game. The student was presented with an equation (e.g. $1 + (2 * 3) + (10 * 10) + 30$) and asked to write the LISP version of this. In addition, the student could see their answer represented as a tree-diagram (achieved by converting the input to a JSON-object, and passing it to the d3.js library).

¹⁷<https://d3js.org/>

7 Others

7.1 Teacher-Student questionnaire system

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- Course: INF116 Object Oriented Programming
- IDE, Editor, Tools: Visual Studio, MySQL Workbench
- Programming language(s): Visual Studio/.NET Visual Basic (VB)

This project was the exam-case in this course, which counted for 60% of the final course grade. The project was based on developing an application for course evaluation. Included in the exam-case was models of the database, and a SQL script that contained some data, with a set user-account (which were to be used for database connection(s)). The system had have two main parts; one for teachers and the other for students. In the teachers part, there was a management section (for questionnaire creation), and a report stage for statistics related to the students answers. The part for the students¹⁸ was mainly targeted at giving access only to students taking the course, and giving them the ability to answer and submit the evaluation. Student information was retrieved electronically from the student system¹⁹.

For each evaluation, a number of questions where made available, where each question should give between two to five answer options. It is up to the teacher to determine which details are relevant for a given evaluation (an evaluation includes the course, evaluation name, created questions and answer alternatives). There were two dates that needed to be set, evalDateOut and evalDateIn, which specifies the time for when the evaluation was available for students. The teacher should be able to change the questions and answers, but not after the first student has completed the evaluation. A course can have multiple evaluations, and this may overlap in time.

For the report section, the teacher should be able to see answer distribution for each question. It should also be possible to know who has completed the evaluation (since some evaluations could be mandatory). There should be an opportunity to "export" the counted data for a specific evaluation to a flat file, which then could be imported into Excel. The teacher should also be able to delete all student responses for an evaluation (when it is finished).

¹⁸ We divided our group into two (we were in total six people), where our "half" worked on the teacher part. Therefore, the main task description for the Student part is omitted from this section.

¹⁹ The student system contained data about which student was taking a given course (where "course" is a subject/course taught a given semester; e.g. INF110 S07). Retrieving student data from the student system was not a part of this project, but just information given in the exam-case description.

7.2 INF165 System Developing Project: Hotel management system

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- IDE, Editor, Tools: Microsoft Access, MySQL Workbench, Microsoft Visio (DataFlow Diagrams)

Storenuten Høgfjellshotel is a mountain hotel with an excellent location in the Norwegian mountains. Aside from their own family-friendly resorts, the hotel has responsibility for maintaining 130 km ski slopes in the surrounding areas. It is a short distance to the municipals newly established state of the art resorts. Høyfjellshotell has since the 50s evolved from a small family business to now being a major employer in the county with people. The hotel is newly renovated and has all the rights.

The hotel has 130 modern double rooms and 35 family rooms with 4 beds, with the possibility of an extra bed. When it comes to stay in double or family, the price includes full board (breakfast, lunch and dinner; beverages for lunch and dinner are not included). The price is per date 1020 kr for an adult per day, and it is half price for children under 16 years. The hotel writes expenses to the room (the guest fills out receipt with room number; expenses can be drinks, mini-bar, etc.), which is then settled at check-out.

After market entry of the ski resort, the mountain hotel has experienced strong growth in its business. They now have an urgent need for a new system to handle their hotel operations. In the first round, it is important that the new system handles the work of managing orders, check-in and stay ("expenses are recognized in the room during the stay") and check-out. On the basis of the information at any time in the system will be possible to take out information of a financial nature.

7.3 INF330 Flash Programming

- Level: Bachelor
- Project type: Group, Assignment
- College/University: Buskerud University College
- IDE, Editor, Tools: Adobe Flash CS 4, FlashDevelop
- Programming language(s): Flash, ActionScript 3.0

7.3.1 Space Invaders

A scroll-game like space invaders, where the player controls a space-ship. The topic for the development was chosen by the group. A short scenario context is that the player is a farmer kidnapped by aliens, and have to save the Earth (and his harvest).

I handled the design and development of the enemies, game settings and playing sound/music. The game contained up to six levels, where the player could select which to play (but level 2-6 had to be opened first by completing previous level. The player got points for destroying enemies (asteroids and aliens). After two minutes of play, the boss appeared. The game also contained a "Top 10" list displaying the scores for the players.

7.3.2 Tower of Knowledge

Tower defence game meets quiz where the goal is to attack only the wrong answers. The topic was selected by the group. The player can put out none or more towers, and the player had to let the "correct" answer only pass.

Answers was shot by clicking the answer, presuming it was within the given towers range. Points were given for attacking enemies, and reduced for each answer passing the goal. The player was not told if the answer was correct until the game had ended, and when the game was over, the player was added to the Top 10 list presuming their score was high enough. My responsibilities in this project was the development of the main menu, reading (questions, answers, Top 10) and writing (Top 10) to file, sound/music.

In addition, I also created a small program for the creation of questions and answers (since the variables was read from file using URLVariables), which was written in Visual Studio/.NET C#.

Example of file content:

qst=question1?,question2?&alt=alt1,alt2&ans=ans1,ans2

7.4 ITE1606 3D-Modelling

- Level: Bachelor
- Project type: Assignment
- College/University: Narvik University College
- IDE, Editor, Tools: 3DS Max
- Repository: <https://github.com/klAndersen/Bachelor-development-projects/tree/master/3D%20Modeling/ITE1606%203Dmodelling>

This course was based on learning how to create 3D-models and how to work with a 3D-modeling tool (3DS Max). This was a purely introductory course, so I cannot claim to have any expertise after this course, other than a general understanding.

Assignment 1-2

In the first two assignments, the task was to create a sketch of a vehicle (I sketched a car), and create textures based on taking and editing images of clothing, carpets, etc.

Assignment 3

The task was to model a Lego block, which should look as realistic as possible. Instead of the text "Lego", our own name was to be added on the one side. It also contained texture, to resemble reality.

Assignment 4

The task was to create a vehicle based on the sketch from Assignment 1. The vehicle was to be modeled in such a way that it could be animated later (in regards to wheels, belts, bone, course change, etc.).

Assignment 5

The task was to model a "race cart/vehicle trail/stadium" for the vehicle to drive on, with tribune and facilities. We also had to take scaling into consideration, in regards to the vehicle.

Assignment 6

The task was to create an avatar model of ourselves, based on pictures of ourselves. The scaling was the same as that for the vehicle in Assignment 4, since the avatar had to "fit" the size of the vehicle.

Assignment 7

Based on the avatar created in Assignment 6, the task was to rig it with a skeleton. This was done using the "bones" and then modifying the "skin", followed by adjusting the "envelope" of the individual members/bones. This was to ensure that the skeleton motion moved the right parts of the figure. The skeleton also had arms and foot-triggers, and had a short animation of a fall by using Inverse Kinematics (IK).

Assignment 8

This task focused on lighting, where the goal was to simulate "evening sun", without using plugins or taking advantage of the "Daylight System". There were also camera usage, that showed a short animation of the vehicle (Assignment 4) driving around on the stadium (Assignment 5).

Index

3DS Max, 27

ActionScript, 26

Adobe Flash CS 4, 26

Android, 13, 16, 17

Android Studio, 16, 17

Arch Linux (OS), 18, 19

Assignment, 4, 9–11, 13–16, 18, 21–28

Bachelor, 3–6, 9, 13, 21, 22, 25–27

BUC, 4, 5, 13, 21, 22, 25, 26

C#, 4, 5, 7, 8, 27

C#/XNA, 6

CSS, 21, 23, 24

Eclipse, 13, 16, 17

Flash, 26

FlashDevelop, 26

Git, 16, 18, 19, 23, 24

GitHub, 18, 19

Group, 18, 23, 25, 26

GUC, 11

HTML, 15, 18, 19, 21, 24

Java, 7, 13, 16, 17

Java Servlets, 13, 15, 23

JavaScript, 18, 21, 24

Master, 3, 11, 16, 18, 19, 23, 24

Microsoft Access, 26

MySQL, 13, 16, 19, 21, 23–26

Notepad++, 24

NTNU, 16, 18, 19, 23, 24

NUC, 6, 9, 13, 27

Oracle, 4

Oracle SQL Developer, 4

PHP, 16, 21, 23, 24

Private, 7, 8, 16, 17

PSPad, 21

PyCharm, 18, 19

Python, 18, 19

Scrum, 18, 23

SQLite, 11, 13, 14

Unity, 24

Visual Basic (VB), 25

Visual Studio, 4, 11, 25

XAMPP, 21

XML, 13, 15–17, 23

XMLEditor, 23

XSD, 23

XSLT, 23