

Richard Huang

Project Specifications Document:

North Arrow



North Arrow

Richard Huang

# Table of Content

[Summary 3](#_Toc495917365)

[Problem 3](#_Toc495917366)

[Goals 4](#_Toc495917367)

[Success Criteria 4](#_Toc495917368)

[Specifications of Program Function 5](#_Toc495917369)

[Academic Record Management Feature 5](#_Toc495917370)

[Undergraduate Programme Feature 5](#_Toc495917371)

[Career Comparison Feature 6](#_Toc495917372)

[Progression View Feature 6](#_Toc495917373)

[User Interface Specifications 6](#_Toc495917374)

[Help Function Specification 7](#_Toc495917375)

[Data Storage Specifications 7](#_Toc495917376)

[Security 7](#_Toc495917377)

[Performance 8](#_Toc495917378)

[Errors 8](#_Toc495917379)

[Software & Hardware Requirements 8](#_Toc495917380)

[Software 8](#_Toc495917381)

[Hardware 8](#_Toc495917382)

[Testing Requirements 9](#_Toc495917383)

# Summary

This program is aimed to assist high school students in finding out details about Undergraduates Programmes they are interested. The program allows authorised users to Input their academic details. Thereafter, the program will generate a list of all possible Undergraduates Programmes in different mainstream universities (currently UCT, UP, Wits and Stellenbosch) in South Africa. The program will allow the see and pinpoint their weakness if there list generated is not long. Authorised User may update their academic details, and the program will produce a new list. All past academic details will be accumulated to the database and can be viewed on a graph tracking academic progress, which also indicates if the students need to put in more effort or not.

## Problem

The problem exists in the inconsistency of APS (Admission Point Score) system, a system of measure in your qualification to enter an Undergrad-Programme, across South Africa Universities. This is often confusing to students and a repetitive walkthrough of the unique algorithm for each university just to measure the possibility of admission. Apart from the problem in the APS system, information on the minimum requirements for each programme is also scattered throughout the long and dull prospectus making it a tedious effort to gather them.

Lastly, Students are often neglecting the importance of academic result, with many saying “a pass is a pass”, and sometime deliberately choosing the “easy” subjects for less effort in academics. Not knowing how that will put a huge disadvantage on students at the start of grade 12 as some of them can’t choose the Programmes they want to do, but instead having to compensate for a Programme that they are not interested in. By providing students hard facts (e.g. a very short list that mostly comprises of humanity programmes) of what exact Programmes they can choose, it informs them what subjects gives them the most choices and what minimum mark is needed at even grade 10, allows students to set aims and plan early.

## Goals

* To allow user to generate all their APS results in an instant, without the tedious work of calculating through complicated algorithms
* To provide a clear view of all possible Undergrad-Programme available, without manually eliminating all the Programmes that requires better academic results
* To accurately trace user’s academic progress and to be presented by a graph, that (hopefully) encourages and warns the user to achieve better results.

Overall to help Gr11, Gr12 students reduce their confusion in the decision making that could possibly affect the rest of their life and for Grade 10s to plan early and choose their subject wisely.

## Success Criteria

The program will be successful if it can:

* Successfully generate the result wanted by the user
* Pop-up table is shown with correct information
* Allow user to access and update their academic records
* Block unauthorised users from accessing the program
* An accurate graph and a suitable warning is displayed

The user-interface will have the following:

* A login/sign up screen that requires username and password OR sign up with specific user information
* A menu frame that allow the user to enter into different features
* A frame that allows user to insert, update their academic detail
* A frame that displays APS result list
* A graph frame where the user can view their academic progress in a graph
* An easy-to-use UI designed using Google’s material design look and feel

# Specifications of Program Function

## Academic Record Management Feature

* **INSERT Academic record**- This feature is accessed when a new user enters their subject choices for the first time or new mark entry inputted and click the “Save” button. This is done on the Student Information frame. User input is archived once the “save” button is clicked then inserted into the database through an **INSERT** SQL query.
* **VIEW Past academic record**- This feature allows the user to review their previously entered Academic records. This is accessed when the “Check your marks” button is clicked. A list of all academic records is displayed in the student information frame. Data is retrieved through **SELECT** SQL queries.
* **UPDATE recent academic record**- This feature allows the user to correct any error or update any mark change or subject changes for any previously entered record. This is accessed when an existing user re-enter mark that had been entered before and clicking the “Save” button This is done on the Student Information frame. Once the Save button is clicked, the new record will be updated into the database. Data is renewed through **UPDATE** SQL queries.

## Undergraduate Programme Feature

This feature will allow the user to view their APS scores for each University programmes alongside its minimum APS requirements, by providing this direct comparison the user can then have a clear understanding of their chances to get admitted. This feature can be accessed by entering the Personalised Programme frame.

## 

## NBT requirement and Filter feature

This feature will allow the user to view which programmes need NBT to get admitted and which doesn’t. This allows the student to decide if NBT is needed or not clearly. This feature can be accessed by entering the Personalised Programme frame. Data is retrieved through **SELECT** SQL queries and derived from the SELECT statement.

The Filter feature act upon the NBT requirement feature. It will eliminate programme that doesn’t match with the filter setting set by the user. this is accessed by clicking the apply button in the Personalised Programme frame.

## Progression View Feature

This feature allows the user to view all their past academic results in a graphical form within the Academic Progress frame. This provide the user with a visual representation of their progress over the years. This shows their strength and weakness in different subjects and encourages them to work towards their goal. It gives a greater impression of the student’s academics ups and downs with its visual aspect. Data is retrieved through **SELECT** SQL queries.

## Programme Search Feature

This allows the user to search a specific programme in the personalised Programme frame. This allows to user to find programmes easily by their title (e.g. a BSc degree), minimum APS needed, and anything related to that record.

# User Interface Specifications

User interface is simplistic and effective. Design and colour will be in a blue and white, combined with material design. UI consist of six frames like mentioned above. A login frame, sign up frame, a main menu frame ,a student info frame, a Programme list frames and a graphic progress frame. UI will be easy to use with help option enabled, this allows the user to learn usage of the program easily and avoid confusion. It will provide all the data that the user desires, and enable back button for the user to return to the main frame or log out to the login frame and switch users.

# Help Function Specification

Within every frame in the program, a help button is present. When the help button is clicked, the help option will display the functions of all other buttons on that frame, a message describing the purpose of that frame and what can be done. The info will assist the user through the usage of the program and provide any extra tertiary education information that needs attention.

# Data Storage Specifications

* A text file "Passwords" will store all registered users' username and password (encrypted)
* A table “tblStudent” in the database will store all information relating to a user’s username, race, subject choices and NBT marks.
* A table "tblYearMark" in the database will include all information relating to a user’s username, grade and subject marks.
* A table "tblTermMark" in the database will include all information relating to a user’s username, grade, Term and subject marks.
* A table “tblProgrammes” in the database will include all information relating to a programme’s name, the university that provides it, the faculty it belong, number of years to complete, minimum APS, subject requirement and NBT requirement.

# Security

All features mentioned above can only be accessed by a registered user’s password and username, while passwords are stored in encrypted form. This avoids unauthorised personal to access user’s private information (academic result) and tamper with the data. This also provides user specific security, thus users can only access their own data.

# Performance

This program will not require any major computer processing resources. Therefore the program should run without lag and respond issues.

# Errors

An error message will be displayed if:

* Incorrect password and username is entered
* Insufficient academic detail is entered
* Academic details out of range e.g. a 110% mark
* Incompatible data is entered in the wrong place (a name in place of a number)

# Software & Hardware Requirements

## Software

* 64 bit Windows OS
* NetBeans IDE 8.2 -Download from: <https://netbeans.org/downloads/>
* Java JDK WindowsX64 –Download from: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.htmln>
* Microsoft Access (2007 or later) is required for database access

## Hardware

Minimum requirements:

* 512 MB of RAM , Intel Core i3 or above, 10 MB on a storage device
* The program needs either a USB port, DVD-ROM or CD-ROM for installation and back up
* A keyboard and mouse will also be required to allow for selection of buttons and inputting of data as well as a monitor to view display.

# Testing Requirements

The program is deemed successful if:

* The program does run and does not crash unexpectedly
* All feature mentioned above work as described
* Security feature blocks out all unauthorised user
* Help option display the corrected message
* Relevant message is displayed in the case of an error
* All data is stored and accessed in the database correctly

A comprehensive debug process will be carried out by inputting correct and incorrect data to successfully test if all features of the program is running as intended.