Note: This is a summary for component 3 of the assessment. Further information related to the marking criteria, regulations and where to get help can be found in the full Assessment brief for

components 1, 2 and 3 of the assessment on Moodle under the section“Assessment Information''.

Component 3: Design, development and testing of a computer

programme solution (Total weight 70%)

Part A- Design documentation (weight 15%)

Overview of banking systems

A banking system is a group or a network of institutions that provide financial services for citizens.These institutions are responsible for operating a payment system, providing loans, taking deposits, performing money transfer and helping with investments. In this coursework, students will create a 'prototype' Banking，system to focus on basic operations such as deposit money, withdraw money, transfer money and agree to make loans to their customers. The system admin can create new bank accounts to customers, search for customers' accounts, update/view customer personal details and close customers bank account.

Use the above activities to draw two flowcharts using the special shapes to represent different types of actions or steps in a process. Link these activities with Lines and arrows show the sequence of the steps, and the relationships among them.For this part of the assessment you need to submit two flow chart diagrams (total weight 15%). To complete this you must refer to Week 2 learning materials.

·(7%) One diagram should depict the log-in process for the banking system

·(8%) Second diagram should explain a critical/complicated process of the system e.g.making money transfer from one account to another.

Important:

This part of the assessment should not be attempted by students until you are provided with the partial implementation code which you must use as a starting point of your system, and this will be provided to you around week 7 of this semester. Developing your banking system from scratch is not acceptable for

this assessment, and may lead into losing substantial amount of marks.

Description:

Rationale

This programming assignment is to apply the programming principles covered in tutorials and lectures to develop a python software that implements core banking solution that is used in banks by its customers and staff members. The software can be fully implemented using text-based interface, however, to achieve higher marks, a GUI based software can be developed using the Python GUI Programming module namely Tkinter. GUI design and Tkinter programming will be covered during week 13. The aim of the exercises is to enhance a student's experience of programming by applying programming principles to a larger problem of developing a complete application.

There are mainly two reasons behind the selection of the banking system as the topic of this coursework. Firstly the students are familiar with the banking systems, hence, students will spend less time and effort to understand the functions specification of the software they will be developing for this coursework. Hence, most of their time will be devoted to the design, development and testing of the banking system by applying the programming knowledge and skills they learnt throughout this module. Secondly, GUI based programs make it easier to interact with the developed system and demonstrate a direct relationship between the user interaction and the system functions and data.

Also, students will learn about how a product works entirely from the user's (or customer's)perspective and not from just a developer prospective. Hence, they will need to develop a user-friendly GUI.

Objective - Sample prototype application

The objective of this assessment is to develop a python software that implements a core banking solution that is used in banks by its customers and bank admins. The provided prototype will include basic banking system functionalities such deposit money and display balance. The main class will allow to add a number of customers accounts at the start of the software. At the initial stage, all customer data will be hard coded. However, for students to achieve higher marks, they should enhance the system functionalities by using text files to store the customers related information.

Assessment specifications with a detailed marking scheme:

To achieve a mark to maximum of 40% of the total marks for the System Development

The application must implement all the following:

Create the necessary classes and functions which allow admins to perform the following tasks

·Admin Login

·Search for a particular customer to perform various banking operations on a customer bank account i.e. check balance, deposit or withdraw money etc.

·Deposit money into a customer account

·Withdraw money from a customer account

·Check current balance for a customer

·View customer details e.g. name and address

·Update customer information e.g. name and address

To achieve a mark of 41% to maximum of 50%

The application must implement all the above and the following:

Create the necessary classes and functions which allow admins to perform the following tasks:

·Close a customer account i.e. remove customer from the system

· Update admin own information i.e. name and address

·Print all customers details

To achieve a mark of 51% to maximum of 70%

The application must implement all the above and the following:

·Customers can have different types of bank account. Accounts will differ in their name, interest rate and overdraft limit etc.

·The bank system should be able to store and load all customers' data from and into a file.

To achieve a mark of 71 % to maximum 80%