TARUMT
TUNKU ABDUL RAHMAN UNIVERSITY OF
MANAGEMENT AND TECHNOLOGY

AMCS1034 Software Development Fundamentals Assignment

Introduction

This is a **group assignment** requiring all students to collaborate in teams; **solo work is not permitted**. Each group must **develop a standalone software application** using **Python** to address a specific problem.

Assignment Specification

Learning Outcomes	Upon completion of the assignment, students should be able to: 1. Translate problem statements into programmable solutions. 2. Develop a software application using basic Python programming concepts.
Assignment Guideline	This assignment consists of three tasks: **Prototype Development**, **Report Writing** and **Presentation**. Please follow the instructions below to complete each task. **<u>Instructions</u>** 1. All students must form groups of **THREE** (**3**) members. Special permission from the course instructor is required for different group sizes. 2. One member should be designated as the **group leader**. 3. Each group is required to develop several applications or features typically found within a **TAR UMT Student Assistant App**. For groups of **THREE** (**3**) members, the program should display a main menu with three applications to choose from. 4. Each member must select one application from the following **10** options, and no two members are allowed to choose the same application: **A. <u>Homework Planner</u>** A Homework Planner helps users create, organize, and prioritize homework to be completed. It allows for setting subjects, deadlines, status and details of the homework. Some advanced features may include categorization, reminders, and collaboration options for team tasks. **B. <u>Simple Reminder App</u>** A Simple Reminder App allows users to set one-time or recurring reminders for class, tasks, appointments, or important events. Users can receive notifications at specified times, ensuring they don’t forget critical responsibilities or deadlines. **C. <u>Basic Calendar/Timetable App</u>** A Basic Calendar/Timetable App enables users to schedule and view class, events, appointments, and meetings. It typically includes features like daily, weekly, and monthly views, color-coded categories, and the ability to invite others to events.

D. Notes Organizer

A Notes Organizer provides a digital space for users to jot down ideas, thoughts, and information. It allows for organizing notes into folders or categories, adding tags for easy searching, and even attaching images or links for comprehensive note-taking.

E. Expense Tracker

An Expense Tracker helps users monitor their spending by allowing them to log expenses, categorize them, and analyse spending habits over time. Some trackers include budgeting features, charts, and the ability to sync with bank accounts for automatic updates.

F. Discussion Room Booking

A Discussion Room Booking allows users to book a discussion for specific times. It typically includes features like selecting a venue (Eg: Library or Cyber Centre), checking venue availability, entering booking details. After booking, the user shall be able to view the booking they have made.

G. Simple Peer Learning Forum

A Peer Learning Forum serves as a platform for students to discuss matters that they faced during study. It allows students to ask questions or share tips with other learners. Users could reply and leave emoji like “Like” to the post.

H. GPA Calculator

A GPA Calculator helps students calculate their Grade Point Average based on their course grades. Users input their grades and credit hours, and the app computes their GPA, often providing options for different grading scales. The app shall also be able to show the student’s academic performance in a chart.

I. Study Pomodoro Timer

A Study Pomodoro Timer is designed to enhance student’s productivity by using the Pomodoro Technique. It encourages users to work in focused intervals (usually 25 minutes) followed by short breaks. The app typically includes customizable time settings and tracking of completed sessions.

J. Flashcard Quizzer

A Flashcard Quizzer allows users to create and study digital flashcards for learning purposes. Users can input questions and answers, and the app often includes features for tracking progress, spaced repetition, and quiz modes to enhance retention of information. Some available quiz modes could be multiple choice questions or open-ended questions.

5. Other applications can be proposed, but they must receive approval from the course instructor. A brief description is provided for each application to guide understanding of their functionality.
6. Customized versions of any of the 10 options can also be proposed, but they are subject to the course instructor’s approval as well.
7. The **Student Assistant program** must be developed using Python only; using any other languages, including a mix with Python, will result in a score of **0%**

marks.

Assessment

#	Deliverable	Marks Allocation
1	Program/Prototype	60%
2	Report	20%
3	Presentation	20%

Please refer to the **Assignment Rubrics** for more information.

Deliverable

All groups are required to submit following **THREE** (**3**) **items** via Google Classroom:

1. **Source code**
   a) Each group member must submit the source code as a single text file for ease of execution.
   b) If the code requires additional configuration for execution, please provide a **README** file with a clear, detail setup description, including a step-by- step instructions, and any necessary libraries or tools.
2. **Report**
   a) The report must be submitted as a **PDF** file.
   b) It should include the following items:
      i. A cover page
      ii. A front page with a plagiarism statement and student signatures
      iii. The main report
      iv. Appendices (if applicable)
   c) The formatting throughout the report must be consistent.
3. **Assignment Rubrics**
   a) Fill in the spreadsheet with the names of all group members.
   b) The assignment rubrics must be submitted as an **Excel** (.xlsx) file.

Submission should be made by the group leader only. If there are three members in the group, the group leader must follow the instructions below to complete the submission:

1. Create a main folder, and name it with the names of all group members in alphabetical order. For instance, if the members are **Tan Li Mei**, **Chan Ai Ling** and **Jeff Wong**, name the folder is **ChanAiLing_JeffWong_TanLiMei**.
2. Place all source code files in a separate folder named **Source Code** within the main folder created in **Step 1**.
3. The main folder should also include other deliverables such as the **Assignment Rubrics**, the **Report**, and the **README** file.
4. Compress the main folder using the Windows Zip function and submit it.

NOTE: The submission of this assignment means that students have agreed that their work is original and comply with the rules and regulations (refer to **Academic Impropriety**).

<table>
<tr>
<td><b>Academic Integrity and Plagiarism</b></td>
<td>Any cheating, attempts to cheat, plagiarism, collusion, or any other actions to gain an unfair advantage in assessments will result in penalties for the involved students. Those found dishonest may face disciplinary action. Therefore, before submitting the assignment, please ensure that all group members comply with <b>TAR UMT Plagiarism Policy</b>.</td>
</tr>
<tr>
<td><b>Submission Deadline</b></td>
<td><b>All groups must submit their deliverables by September 14, 2025 (Sunday of Week 12), before 12:00am.</b>

Late submission will be penalized:
<table>
<tr><th>#</th><th>Late Submission</th><th>Marks Deduction</th></tr>
<tr><td>1</td><td>Within 1 - 3 days</td><td>10%</td></tr>
<tr><td>2</td><td>Within 4 - 7 days</td><td>20%</td></tr>
<tr><td>3</td><td>After 7 days</td><td>100% (Coursework will be rejected)</td></tr>
</table>
Failure to submit the deliverable will result in a failure of the coursework.</td>
</tr>
</table>

Report Writing Guidelines

Section	Name	Content
A	System Description	System descriptions are simplified documents that can provide end-users with a clear overview of a software system's features, and functionalities. Write a concise system description that provides an overview of the system, detailing its purpose and features, accompanied by screenshots of the GUI for each feature.
B	Stepwise Refinement	Stepwise refinement is a systematic approach in software development and problem-solving that involves breaking down a complex problem into smaller, more manageable parts. Use diagrams for illustration to refine your solution by structuring the problem into smaller components.
C	User Manual	User manuals are comprehensive documents designed to guide users in effectively using a product or software application. Write user manuals that include clear, step-by-step directions and visual aids, making it easier for users to understand and utilize the personal assistant application.
D	Test Cases	Test cases serve as a blueprint for testing, ensuring all functionalities are examined and documented to identify bugs or issues before deployment. Design test cases that specify inputs and expected results to validate various aspects of the application.