* **Final Group Report [14%] & S/W [6%]** Submission Deadline: Apr 30 (Sun), 2017 at 23:59.

Total : 20% of CA

* **(Final Report + S/W) [in .zip] should be submitted via the course Moodle webpage at the HKU portal.**

This group project specification contains the following items:

* For each of the above projects, in the final report, each group should submit the finalized design and implementation of the Java/Swing OR a mobile application. The mobile application can be either the Android, Python/Kivi or Apple iOS (if the group is more familiar with the iOS platform) application.

RULES:

- ✓ ***Grouping***: To encourage the spirit of teamwork in industry, each group should consist of 2 (max.) members. Each member should take up equal share of work, and be actively involved throughout the analysis, design, implementation, testing and documentation phase. Any member who is **not involved in any part of the group project will immediately receive zero mark for that part** as clearly defined in this project specification. Any student who fails or feels reluctant to pair up with another student can form a single-person team who can choose freely to work on any of the 2 suggested projects. However, there will be **no extra bonus to be given to such single-person group though the concerned student can be allowed to work a reduced scope after seeking the approval from Dr. Vincent Tam.** Each submission should be evaluated primarily on the "quality" of the work. Besides, **no group is allowed to have more than 2 students for fairness unless with our prior permission**!!

- ✓ ***Submission***: For the preliminary/final report, each group is required to submit only ONE single copy. For the final individual report, all the source programs/files (such as .html, .java, etc.) required to **successfully build** the ultimate graphical user interface (GUI) should be included in the ZIP file. Otherwise, no mark will be given to the S/W part. To **ensure standardization and fairness** across the different group projects, students are restricted to use the Java/Swing Programming to implement their designed GUI of the desktop app. For the mobile app, students can use either the Android SDK, the Apple Xcode IDE or Python/Kivi framework. If any group uses any publicly available Java/Swing library downloaded from the Internet or other source, they should duly acknowledge/declare the source(s) in their final report; otherwise, NO mark will be given to S/W part.

To standardize the evaluation strategies of different projects, the SAME evaluation criteria were SET for EACH group project as follows:

Evaluation Criteria	Marks
ii) **Clarity and Presentation of the Final Group Report** (i.e. whether the report can clearly present any modification from the initial design to final design, justify the reasons behind such modification, and the detailed final design [including the interaction style(s), user models, state diagram(s), the logic-flow (control) diagram, event handling mechanisms, data-flow diagram, and the detailed human/technical factors considered, and all the main findings including the weakness and strengths of the implemented system/interface on the desktop app versus those of the mobile app]?)	8%
iii) **Clarity of the S/W Documentation = Source Programs** (i.e. whether the source programs contain sufficient and clear comments ?)	3%
iv) The Implementation/Software: **a) Basic Functionalities of the submitted S/W for desktop or mobile application** (i.e. whether the S/W satisfies the system specifications stated under each topic?)	6%

b) Clarity and Presentation of the submitted Test-and-Evaluation Plan (i.e. whether the test plan is concise and clear enough to direct an effective testing on the "basic functionalities"? whether the evaluation plan is carefully planned and detailed enough to consider most average and exceptional cases?)	3% Sub-total = 20%

- Each group will be roughly given around 8 weeks (now to Apr 30, 2017) to design, implement and prepare documents for the chosen group project. Below is our brief "suggested" timetable of 10 weeks for each team to manage his/her own group project:

Suggested Activities	Duration
a) Analysis & Design	2.5 weeks
b) Implementation	3 weeks
c) Testing and Improvement	2.5 weeks
d) Preparing Documents for Group Submission	2 weeks

** Students are gently reminded that each group project should NOT be focused on as a "programming exercise" as obvious from the above mark allocations that only 6% (out of 20% in total) are directly related to the basic functionalities of the S/W produced. During marking, we would carefully evaluate on the problem analysis, design principle and practical issues (such as human factors) considered, and the completeness of the evaluation plan. Therefore, to ensure fairness, we may ONLY provide some relevant reference material on Java/Swing or Android programming, and may NOT be able to provide "any specific technical support" for each individual student/group due to our limited manpower.

A Checklist of Items Required for the Submission of Each Group

** Each of the following submission items should be identified with each student's FULL NAME & UNIVERSITY NUMBER printed on the cover page of the concerned document.

Items for Submission	**Details**
i) **Executable Files (.class and all required files such as HTML…etc) + Source Programs (if any, like xx.java for Java Swing programs) for the implemented S/W by the group**	If extra libraries are required for compilation/execution, please include in your ZIP file. Also, for some specific compilation/execution setting, please include an"README.txt" to clearly explain.
ii) **The Final Group Report** (at least 8 pages per group excluding the program listings) which should at least contain the following discussion: - Problem Definition (if any); - Basic Functionalities of the Responsible Sub-System; - Design Approach/Principles Used; - The Targeted USER GROUP; - (Human/Environment) Factors Considered; - Strength and Shortcomings of Existing Sub-System; - Future Extensions/Improvements (if more time is given). -	For clarity, each report should have a "Table of Content" after the cover page. Also, each page of the report should be numbered on the lower RHS corner.
iii) **User Manual** (at least 1 page) to explain to the "targeted user groups" how to use the basic/extra functionalities of the implemented sub-system or functions **+ ALL Program Listings** (no page limit !) for each Group submission	The user manual should be clearly written for the targeted user groups to learn to use the basic functionalities.
iv) **A Separate Test-and-Evaluation Plan** (at least 1 page) to clearly explain to marker how to TEST and EVALUATE the concerned functions/sub-system for each Group submission. Accordingly, it should contain at least two parts: a. TEST PLAN – how to use a step-by-step approach to test all the basic functionalities of the implemented sub-system satisfied the system specification stated in this **specification**; You should also include the average and exceptional cases for testing in this part! b. EVALUATION PLAN – In case you're allowed to send to different users for evaluation, how would you set up the evaluation scheme (e.g. questionnaires or on-site evaluation, etc) and what evaluation criteria you should	

use to determine the success/effectiveness of your implemented sub-system.	

** : item i), ii), iii) and iv) to be compressed into ***1 single ZIP file named "Grp-XX -Final-Report.ZIP"*** for submission via the course Moodle webpage at the HKU portal by Apr 30 (Sun), 2017 at 23:59, where XX is the assigned group number.

Project 1: A Smart Living Application

The next-generation smart living application includes the control of many smart home appliances including smart TV that will integrate light/temperature/touch/sound or image sensors to sense its environment ***and*** green and healthy living styles like suggestion of meals with low calories intake and daily exercises. In this project, each group is asked to design and implement a smart living application with interactive user interfaces to control at least 2 (i.e. min = 2) different types of smart home appliances like smart vacuum cleaner and smart washing machines AND suggest at least 1 type of service like green eating or daily exercises suitable for each individual's needs. A smart living application should consist of a navigation menu for end users to choose the types of smart home appliances to control OR the styles of smart eating/exercises to recommend. In the specific user interface for a particular smart home appliance, the users should be provided with clear and interactive interface objects / controls to control the underlying appliance, e.g. a maneuver/directional menu to control the movement of a smart vacuum cleaner. The implemented user interface does not need to be actually controlling a genuine home appliance in a real-time manner. Specifically for the smart home control in this project, it is essentially *a 'simulated' user interface* that can be potentially used to control the operations of the designated smart home appliance. Thus, there is no need to turn on any wireless network to send out the control commands. After a user clicks or selects a GUI object on this control interface, the application may simply display a pop-up window or label to notify the user about the result or status of the operation, e.g. "The smart vacuum cleaner has successfully completed its operation.". In addition, the smart living application will compute and store the accumulated duration (hours & minutes) of operating each type of smart home appliances in a local text file or database. With some predefined power consumption rate for each type of home appliance in the program, the application can be used to generate a simple report of power consumption for the concerned smart home appliances in text labels or a pop-up window. Below is an example of the power consumption report.

Power Consumption (Until : Feb 26, 2017)

Smart Air Conditioner – Usage : 3 hrs 20 min, Power Consumption : 0.61 kW
Smart Vacuum Cleaner – Usage : 1 hrs 10 min, Power Consumption : 0.53 kW

The medium of instruction must include English to cater for the global market. Other than English, you can add any additional language such as Chinese as appropriate. You can provide a simple login screen for registered users to log in and then choose the appropriate control screen to try to control the underlying appliances or recommend certain aspects of smart living styles. The operations performed by each registered user can be stored in local databases, of any form like a simple text file, excel file or relational database file, on the desktop computers or mobile devices. Besides the basic functions of login, the system should clearly display the newly performed operations and possibly the current [energy/positional] state(s) of the involved smart home appliance or the current health condition of the concerned smart living users. The aim of this group project is to allow student practising the design principle and the JFC/Swing or mobile app dev. knowledge acquired in this course and help the students to appreciate the flexibility of programming-with-libraries approach for building useful GUI and OO-based application for the future smart living environment.

Below are some suggested links for reference on relevant topics – DO NOT copy from such or any other links:

http://smartliving.hkt.com/eng/

http://www.erb.org/smartliving/en/

http://www.expertreviews.co.uk/general/1301596/samsung-smart-home-appliances-are-the-future

http://www.samsung.com/us/home-appliances/

http://www.techhive.com/article/2024393/smart-home-appliances-are-a-big-deal-at-ces-2013.html

http://www.greentechmedia.com/articles/read/smart-homes-meet-real-world

Project 2: An e-Book Application with Quizzes for Primary School Kids

No matter on desktop computers or mobile devices, the huge demand for e-books and educational games is growing extremely fast in the past few years. And the trend of growth will likely continue in the coming years. However, carefully designed e-book bundled with educational game-based quizzes for primary school kids are often ignored in many places including Hong Kong. In this project, you should design an e-book application with game-based (i.e. fun to play and learn) game-based quizzes for primary school kids. The medium of instructions is English to cater for both the Asian and global markets. In general, the targeted end-users fall in the age range of 6 ~ 12 years old for school kids. In case you do not agree, you can specify your own targeted age range, say from 12 ~ 16 in your design. It should be noted that educational games differ from general computer games for entertainment that they should have an educational purpose, like helping the teenagers to learn about some important concepts in Science or Maths, English or Music, pronounce or spell a word for learning languages (Chinese or English, etc.), etc. as set according to the relevant content/topics of the e-book materials. The rules of the game should be easy-to-understand as according to the properties of the different age groups and the level / type of skills or knowledge the kids already acquired. After all, sounds and colors should be used whenever appropriate to attract the teen's attention. The software should have a carefully designed and easy-to-understand introduction screen with several options (as menu items or check-buttons) to open / close the reading of the reading e-book materials (in PDF or other common format), and followed by some different levels of difficulty (easy, intermediate, most difficult, etc.) for the kids to select and play around with in the game-based quizzes. The detail of different parts of the quiz game should be clearly explained via the online help facility. The prototype

implementation of this project is completely up to the individual group. No extra bonus will be given to any networked educational quiz game.

Below are some suggested links for reference on e-books and educational quiz games – Pls. DO NOT copy your game from such or any other links:

http://www.ebooks.com/

http://www.free-ebooks.net/

http://en.wikipedia.org/wiki/E-book

http://www.neok12.com/games.htm

http://www.learninggamesforkids.com/animal_and_nature_games/randomals/animal-quiz.html

http://quizzes.familyeducation.com/

Project 3: A Data Analytics Software for Computational Finance

In the area of computational finance or financial engineering, many analysts, financial engineers or investors will employ different software tools to perform critical analysis on the stock data such as the stock prices including the current price, the daily open/closing price, the daily high/low price, and/or volumes, etc. On top of it, there are various technical indicators such as the 10-day, 20-day and 50-day moving averages (MA's), relative strength indicator (RSI), the Bollinger bands, etc. to gauges / estimate the relative performance of different stocks in the stock market. Due to the volatility and fluctuations of the local / global stock markets in recent years, the uses of data analytics software to analyze on the performance of a particular stock or fund in the stock market is very popular, especially in the world's financial centers like Hong Kong, London, New York and Tokyo.

In this project, you should design a data analytics application with various (as least 5) technical indicators available for the end users to choose. The medium of instructions is English to cater for both the Asian and global markets. In general, the targeted end-users fall in the age range of 18+ years old. In case you do not agree, you can specify your own targeted age range, say from 25 in your design. Besides, you may specify your targeted end users as the general users (i.e. anyone from the general public), financial analysts or fund house managers. Depending on their specific user type, you should clearly define their computer skills and domain knowledge. The data analytics application should be easy-to-use as according to the properties of the different age groups and the level / type of skills or knowledge the end user already acquired. After all, sounds and colors should be used whenever appropriate to attract the end user's attention. There is NO need to continuously download the real-time & online stock data for analyses. All the relevant stock data for analysis in the application can be pre-stored as some offline data in a file, such as a text, excel or database file, of the desktop computer or mobile device. The software tool should have a carefully designed and easy-to-understand introduction screen with several options (as menu items or check-buttons) to select and display the

corresponding technical indicators. Charting facility (for plotting line-charts, bar-charts or pie-charts, etc.) is not strictly required yet it may be provided as some additional feature to enhance the data analysis. The detail of different parts of the software tools should be clearly explained via the online help facility. The prototype implementation of this project is completely up to the individual group. No extra bonus will be given to any networked software tool.

Below are some suggested links for reference on analytics software or technical indicators for computational finance – Pls. DO NOT copy your application from such or any other links:

https://en.wikipedia.org/wiki/Technical_indicator

http://www.investopedia.com/terms/t/technicalindicator.asp

http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_bands

http://www.investopedia.com/articles/active-trading/121014/best-technical-analysis-trading-software.asp

https://www.metatrader4.com/en

http://www.ges.com.hk/en-us/

----END OF Group project Specification-------