

CSCI 524 Assignment 03
Case Study and System Vision Document
100 points+10 bonus points

Instruction:

- Show your work (at least 50% penalty otherwise)
- Submit **a single WORD document (*.doc or *.docx only)** containing all your answers to the assignment folder ("Assignment 1") under D2L (at least 10% penalty otherwise)
- Make sure you submit the intended one. You should download what has been uploaded and double-check if the correct document has been submitted.
- You can submit as many times as you want, but the last submission will only be graded. If the last submission is made after the deadline, there will be a late submission penalty.
- No resubmission/extension request will be accepted after the due date.
- No plagiarism:
 - Do not copy and paste any from textbooks and other resources to answer questions (Zero points will be given otherwise).
 - Answer the following questions in your own words based on the concepts and contents that we have studied in the lectures.
 - Turnitin file similarity should be less than 10%
- AI misuse or AI writing detection (<https://www.turnitin.com/solutions/topics/ai-writing/>)
 - Should be less than 10%

Questions

Qn1. (20 points, 2 points each) Match the key terms below with the definition that best fit them.

Complete system tests and then deploy the solution	Interface	Design the system components that solve the problem
Prototypes	Plan and monitor the project	Discover and understand the details
systems development methodology	Application software	Systems Analyst
Build, test, and integrate system components	Domain Modeling	Identify the problem and obtain
Agile Methodologies	System	Component
SDLC	Repository	Rapid application development
Boundary	Use cases	Environment

- 1) _____ The first phase of the SDLC in which an organization's total information system needs are analyzed and arranged, and in which a potential information systems project is identified and gotten approved.

- 2) _____ In software engineering, the framework identifies all the activities required to research, build, deploy, and often maintain an information system. There are several processes required for a high-quality information system.
- 3) _____ Software designed to process data and support users in an organization. Examples include spreadsheets, word processors, and database management systems.
- 4) A series of _____ is built and adjusted to meet user requirements.
- 5) _____ Newest development technique used for system development that focuses on adaptive methodologies, people instead of roles, and an overall incremental, iterative development process.
- 6) _____ The organizational role most responsible for the analysis and design of information systems.
- 7) _____ Phase of the SDLC, in which preliminary fact-finding is conducted to understand requirements and details
- 8) _____ Phase of the SDLC, in which the system's high-level structure is designed and user experience and database are defined.
- 9) _____ A standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.
- 10) _____ Techniques for identifying and modeling domain classes for systems analysis

Qn2. Case Study (40 points)

Geocaching System When Wayne Johansen turned 16, his dad bought him a new Garmin handheld GPS system. His family had always enjoyed camping and hiking, and Wayne usually monitored their hikes with his dad's GPS system. He always liked to carry the GPS to monitor their hikes' routes, distances, and altitudes. More recently, though, he had found a new hobby using his GPS system: geocaching. Geocaching is a high-tech version of the treasure hunts that most of us did as kids. Participants search for geocaches or caches that are small, hidden, waterproof containers that typically contain a logbook and perhaps a small item. When found, the participant sometimes gets instructions for the next move—entering information into a logbook or looking for the next cache.

As Wayne became more involved with his hobby, he discovered many activities for geocaching enthusiasts. The simplest ones involve caches found using GPS coordinates, although even some of these tasks can be difficult if the caches are well hidden. Some of the activities involve multipoint drops where there is a set of clues in multiple locations that must be followed to arrive at the final cache point. Some activities include puzzles that must be solved to determine the coordinates of the final cache.

Before long, Wayne wanted to make his own caches and post them for people to find. He discovered several Web sites with access to geocaching information, caches, and memberships. He joined one of the geocaching Web sites and used it to log his findings. However, he decided he would like to create his own

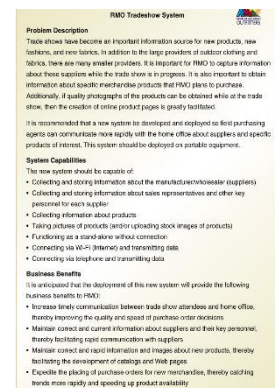
system to track all the information he had about his caches. Conveniently, Wayne's older brother Nick, a college student majoring in information systems, was looking for a semester project for one of his programming classes. The two decided to develop a system to help Wayne keep track of all his geocaching activities.

In this end-of-chapter case, you will go through the various core processes of an SDLC and perform some of the development project activities. The project is divided into days, as was our Tradeshow System project. This case's daily assignments should be considered preliminary efforts and rough drafts. These assignments aim to help you remember the overall approach to software development. Several assignments have been listed for each day to allow your instructor to select those that best meet the course's objectives.

Brainstorm all the functions this geocaching system might do and write a rough draft of the System Vision Document based on your brainstorming Ideas (as our Tradeshow System project has been studied)

Hint:

- Think of what Wayne wants the system to do and why this helps him.
- Keep it at a very high level. These activities closely relate to Core Process 1: Identify the problem and obtain approval.
- The document needs to include three sections: **problem description**, **system capabilities**, and **business benefits**. One example can be Figure 1-08 in 01-csci524_lecture_Intro.pdf.



Formatting requirements:

- Standard size (8.5 x 11" in the US), Page Margins (1" on all sides), Font (12 pt., Times New Roman), 1.15 spacing
- No page limits. 1 or 2 pages are long enough for this case study.

Q3. Programming Practice (40 points)

Problem Statement: Design a Geocaching System in Java that allows users to:

- Register and log in with a username and email.
- Add geocaches with details such as title, description, coordinates, and difficulty level.
- Search for nearby caches based on a given location.
- Log a found cache, adding a timestamp and optional comments.
- Track user statistics, such as total caches found.

Based on the above, 1) (20 points) design a UML class diagram (using tools like Lucidchart, Draw.io, or StarUML) that represents the system's structure. Two guiding questions are a) "What classes will you need?" And b) what attributes and methods should each class have?"

2) (20 points) Implement the above design in Java with a menu-driven interface (an example is available below.)

```
// Menu Interface Example:  
Welcome to the Geocaching System!  
1. Register User  
2. Add Geocache  
3. Search for Caches  
4. Log a Found Cache  
5. View User Stats  
6. Exit
```

An example of a menu-driven interface in Java can be found in the file “MenuDrivenProgram.java”. Your implementation should follow Object-Oriented Design principles and include appropriate classes with attributes and methods.

Bonus Features (10 points-Optional):

- Include two more major functions/classes listed in Q2 in the UML diagram and implement them in Q3.
- Store data in files for persistence.

What to submit:

- Your answers to Qn1, Qn2, and Qn3-1) in a single .doc or .docx document, ensuring that the similarity score does not exceed 10%
 - your system vision document should be included as your answer to Qn2
 - Please ensure that both the answers to Q1 and the system vision document are in the same Word file.
- Copy the code from .java files and paste it into the same Word file.
- Show and enclose the outputs of the project in the same word file.
- Enclose the UML diagram in the same word file.
- Submit the Word file, UML file, and all .java files electrically through D2L.