# theUnlimited – Assessment

Thabang Mogano

27 / May / 2022

## Part One – Tourist / Knapsack / Problem

1. What did you find challenging about this problem?

   The first challenge that I came across was the algorithm itself, as it posed more questions than answers for me.  This is because I had to make a few assumptions:

   - How to work out the importance of an item that has to go into the knapsack based on a combination of two (2) values i.e. its weight and value.  In this respect, it made it a combination problem. This means that with every item in the collection and at every step, what is the best and/or optimal combination of items that can go into the knapsack.
   - With every step in the computation it posed challenges in terms of items being constantly added / removed / replaced between two (2) lists i.e. the list of predefined knapsack items and those selected into the contents of the knapsack.  I ran into a StackOverFlowException, of which puzzled me for a while. The pseudo-code followed:
     - At every step
       - An item is added to the knapsack based on the weight and value
       - If the set the would no longer be feasible
       - Reject items under consideration (and never considered again)
         - Else if, set is still feasible, Then
           - Add the current item

2. What design/development principles/patterns did you use in your solution?
   - In order to solution for this problem, I considered the following:
     - Using interfaces, so as to avoid inheritance.
     - Using Composition, so as to also avoid inheritance.
     - By using Composition and interfaces it allowed me to follow closely the rules of the following solid principle(s):
       - LSP (Liskov Substitution Principle) – using a derived class instead of the parent class
       - OCP (Open/Closed Principle) – a class should be open for extension but closed for modification
       - ISP (Interface Segregation Principle) – classes / clients should not be forced to implement interfaces they do not need
       - Dependency injection

3. How long did it take you to complete the solution?
   It took me about a day and a half to solution for this problem. My time was split between analysis, testing and development. The decision to use either a console app or web app or winforms app was only considered once the base libraries were put into place.

# Part Two – Project / Werewolf

1. After looking at Warewolf, what skills have you got that you think would help with the development of the application?

   I believe that I bring the following soft and technical skills which I have honed over by 10+ years as a developer in various industries and positions from junior to intermediate to senior developer in order to further the development of the Werewolf solution.

## Soft skills

- Problem-solving skills and critical thinking.
- Communication skills.
- People and interpersonal skills.
- Self-awareness.
- Self-learning.
- Accountability.
- Time management.
- Emotional intelligence.

## Technical skills

- Programming Design Paradigms / Patterns
- Requirements Gathering
- Business and Technical Analysis
- Data Modelling and Database Design
- System Architecture and Design
- System Documentation
- System Unit Testing
- Agile Methodologies
- DevOps Methodologies
- SRE Methodologies
- SOA (Service Oriented Architecture)
- 12 Factor App / Microservices

- CSharp (C#), Visual Basic, VBA, Java
- MS Visual Studio, Eclipse, NetBeans
- MS SQL, MySQL, DB2, Entity Framework, MongoDB
- SSIS, SSRS, Crystal Reports, Business Objects
- WCF, Web Services, Rest Services, MVC
- ASP, ASP.Net, IIS, Angular, NodeJS, ExpressJS
- HTML, CSS, JavaScript, jQuery, AJAX, TypeScript, PHP
- FTP, SFTP, WinSCP, ProtoBuf
- ArcGIS, ArcView, ArcInfo, ArcMap, TNTMips
- Tortoise SVN, Perforce, MS SourceSafe, TeamCity, GIT, SourceTree
- JIRA, Confluence, UML, draw.io
- Autosys, VisualCron, Windows Task Scheduler

2. Why do you want to work on this product?

   I love the product name; it is unique and interesting. I want to work on this product because of its open source capabilities and nature as I believe in collaboration and sharing knowledge. The product branching strategy is very interesting to me, as I have never come across quite something like it before. I have mostly worked on monolithic projects that have posed major issues when trying to apply new technologies and agile methodologies.  I believe the way that the product is created and setup will open me up to more ways of thinking, planning, sharing and learning.

3. What do you not like about the product?
The open source nature of the product may also be a bit of a drawback and pose some challenges but as mentioned in my answer and reflection in question two (2) it helps foster collaboration and the sharing of ideas. How are gatekeepers selected for the product? How are issues of code conflicts solved? How quickly are those solved, if they arise?