Statement and Confirmation of Own Work

A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.

Student Declaration

I confirm the following details:

Candidate Name:	YANKHO KAMPHANDULE
Candidate ID Number:	P00202451
Qualification:	L5DC
Unit:	AGILE DEVELOPMENT
Centre:	NACIT(BLANTYRE)

I have read and understood both NCC Education's *Academic Misconduct Policy* and the *Referencing and Bibliographies* document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.

I confirm that this is my own work and that I have not colluded or plagiarised any part of it.

Candidate Signature:	[illegible]
Date:	09 APRIL, 2025

TABLE OF CONTENTS

TASK 1: Introducing DSDM and Agile practices. 3
TASK 2: DSDM Principals 5
TASK 3: DSDM Products 7
TASK 4: Requirements/User Stories 8
TASK 5: Modelling (Sequence Diagram, Class Diagram) 10
TASK 6: Prototyping 11
TASK 7: Retrospective 13
REFERENCES 17

TASK 1: Introducing DSDM and Agile practices.

a) DSDM is an efficient and effective method used when developing a system, it involves continuous iterative development and maximal user involvement, below are advantages why the use of DSDM is a great choice for Geekup compared to the traditional waterfall approach referencing each stage of the process.

i. **Early clarity**
The pre-project provides Early clarity, the business case is defined and makes sure the project is worth pursuing before resources are spent, while the Waterfall model simply jumps straight into requirement gathering stage without evaluating the project overall value. In context of the scenario, Disrupt Digital will have sat down with Sarah and Billy Hampton to clearly talk about the value of the project before any developing is started.

ii. **Guaranteed Stakeholder alignment**
Because there is clarity provided early on, the DSDM process guarantees Stakeholder alignment, this will engage the business users from the very start, while the Waterfall Model just focuses on documentation before involving key stakeholders, this leads to misalignment
of requirements.

iii. **Focused and short**
The DSDM process is Focused and short because it timeboxes the feasibility study in order to quickly determine whether the project is feasible while the traditional Waterfall model involves long planning phases that sometimes become outdated before development begins. In context to the scenario Digital Disrupt will make sure to finish the feasibility study in around one month to still have time to finish the project in the six-month period.

iv. **Business-value driven**
This means its requirements rely on business goals. DSDM prioritizes solutions that offer business benefits, while the Waterfall model concentrates on technical details early on, which usually doesn’t directly support business goals.

v. **Active user involvement**
This is significant in the DSDM process. It includes users to understand real-world business processes, while Waterfall relies on general analysts to interpret user needs, this leads to misinterpretations. In context to the scenario Digital Disrupt will involve Billy and Sarah at every stage of the process to check if they are satisfied with the progress.

vi. **Iterative improvement**
Iterative improvement makes the DSDM process the ideal choice as it allows adjustments based on business requirements during the study. Waterfall finalizes requirements early, making it costly to make changes later.

vii. **Prototyping**
DSDM builds and improves prototypes depending on user feedback while the waterfall model designs everything upfront, which might lead to a system that doesn't meet user expectations. Digital Disrupt will provide Geekup with a work version of the application at every stage.

viii. **Early error detection**
DSDM provides early error detection because it has an iterative approach that identifies design flaws early, while in Waterfall errors are noticed during testing and fixing them is costly.

ix. **Continuous delivery**
In DSDM continuous delivery allows parts of the system to be deployed for real use while Waterfall requires waiting until the very end, this delays feedback.
The overall development workload because users are involved and testing happens throughout, DSDM products are more refined by release. Waterfall systems often need significant changes after delivery. (Scholz, 2025), (Richards, 2023), (Messenger, The Dsdm Atern Student Workbook a Guide to the definitive agile framework, 2012), (Lyoshin, 2025), (Venema, 2024)

b) Below are the main phases of the DSDM lifecycle, together a brief description on each phase:

i. **Pre-project**
This phase ensures the project is worth pursuing and properly set up before work begins. At this stage the business needs are Identified, initial funding and resources are allocated, the project team members are appointed tasks.
All in all the high-level scope and objectives are identified and the Project is approved only if it purposeful after that the whole project infrastructure and responsibilities are established.
Billy and Sarah sit down to see if the application is a really a good step or if it will just end up failing and wasting their resources

ii. **Feasibility**
This is the stage where the projected is analysed to determine if it is technically and economically viable. Business problems and objectives are assessed, benefits and risks are identified, staff determine if DSDM is suitable for the project, after all the required analysis a feasibility report is produced and an outline plan for the next phase is established.

Disrupt Digital have a meeting with Billy and Sarah to discuss whether the funding of £25,000 will be enough for the full development process, also they discuss Billy and Sarah's Objectives(the project must be finished in six months time).

iii. **Foundations**
This is the phase where the business processes and user needs are understood the in detail, business processes and data flow is analysed, key users are identified and the roles they play, all high-level requirements are defined at this stage, after all that is finished a prioritized list of system functions is produced. Understanding functionalities like the user being able to search 3 times before being prompted to create an account (on the user side) and understanding that the app will only be an enterprise application is done at this stage.

iv. **Exploration**
At this stage a working prototype is developed, it demonstrates how the system will function the prototype is developed based on requirements this help gather user feedback, if user has any problems with the prototype, it is refined and brought back for another evaluation, requirements are updated since user were was changed after using prototype. Digital disrupt will provide the login page to Billy and Sarah to see if they are satisfied with the way it functions.

v. **Engineering**
This phase focuses on turning the functional models into a fully working system incrementally. The design is refined and development the actual system, continuous testing is performed, feedback is and improve, system components are tested, the documentation is updated, designs for system are finalized.

vi. **Deployment**
At this phase the system is deployed into the live business environment final system integration and user acceptance testing is performed, end-users are trained, the system is put into operation. A fully working system is introduced to the production environment, user manuals and training materials are published, this is where the post-implementation review occurs.

vii. **Post-Project**
This phase evaluates the success of the project and support of the system after release. At this stage review lessons are learned, all benefits are evaluated, ongoing support and maintenance is provided. A post-implementation review report is published together with a support documentation. (Agile Business, 2025), (Messenger, The DSDM Atern student workbook a Guide to the definitive agile framework, 2012), (Janse, 2025) (Agile Business Consortium, 2025)

TASK 2: DSDM Principals

Below are 5 principles of DSDM and how they would affect the scenario:

i. **Focus on the Business Need**
In DSDM every decision made throughout the project timeline is focused on delivering real business value. This principle ensures that the team never loses sight of why the project is being undertaken in the first place. Before any technical work begins a clear business case is defined, it outlines what the organization hopes to gain from the system being developed. The business case guides the whole project and if at any point it is observed that the business value can no longer be delivered, the project can be stopped or re-scoped.
In context to the scenario, Digital Disrupt will make sure the application being produced has a strong feature to validate authenticity of any product that wants to be displayed on the application, also they have to include features that allow users to share their location for delivery purposes since the business is an application only enterprise.

ii. **Deliver on Time**
This principle emphasizes on delivering results within a fixed timeframe. This is achieved by using timeboxing (this is where tasks are given a certain period of time and have to be completed before the time runs out). If time runs out, lower-priority requirements are dropped or postponed, but the deadline remains the same.
In context to the scenario, the project has a timeframe of six months, this means each task or phase will be given a certain amount of time to be completed, if it happens that time is about to run out and a certain task is not completed Digital Disrupt will consult Billy and Sarah to find out whether they can postpone it or discontinue it. (Dynamic Systems Development Method (DSDM): Driving Agile Project Success, 2023),

iii. **Develop Incrementally**
DSDM encourages building and delivering the system piece by piece rather than building the whole system in one go. This development approach allows for early delivery of usable parts of the system and frequent feedback from users. Each increment adds value and is tested and reviewed before moving on.
In context to the scenario, the sign-in and sign-up page will be delivered to Billy and Sarah to get their feedback on how they feel about the design and if the functionality meets their standards, if it does Digital Disrupt will go on to develop the next piece of the system, If they don’t like it Digital Disrupt takes it back and they refine it depending on the feedback they got from Billy and Sarah.

iv. **Collaborate**
Collaboration promotes close cooperation between developers, business users, and other stakeholders from start to finish. This ongoing involvement ensures that the solution being developed truly reflects the needs of the business. This ensures the solution meets both the business and functional requirements. Regular reviews are used to facilitate collaboration and maintain alignment.
In context to the scenario, Digital disrupt will make sure Billy, Sarah and the staff are all on the same page at all stages of the development process, developers at Digital Disrupt can have a say on certain features of the application, Billy and Sarah will allow the staff to enlighten them since they are the professionals.

v. **Communicate Continuously and Clearly**
This is a very important factor in DSDM as it keeps everyone informed and it resolve issues quickly. This principal encourages an open dialogue, face-to-face interactions, and the use of

visual aids like charts or task boards to share information.
In context to the scenario, Digital Disrupt will go to Billy and Sarah if any issues come up while developing the application, even if no issues occur then they will still communicate all progress by producing models and prototypes, this decreases risk of the final product not meeting user requirements. (Zartis Team, 2025), (DSDM: Definition, Benefits, Principles and Practices, 2025), (Dynamic Systems Development Method (DSDM), 2025)

TASK 3: DSDM Products

Below are five DSDM products that could be generated during the duration of the application development together with a short description of each product and which role it is involved in.

i. **Terms of Reference**
This is produced at the Pre-project Phase of the DSDM process, it’s a high-level statement of the business problem or opportunity and the scope of the proposed application. For it to be approved stakeholders must agree that the project idea aligns with strategic goals and is worth investigating further. Approval is based on initial viability and business relevance. This is a Milestone product. In context of the scenario, it could be a document outlining the problems that Billy and Sarah want to overcome with the application, any opportunities like how it will be a free-to-download app and scope (e.g. it will be produced for the Android platform then iOS and that it will be an enterprise app).

ii. **Feasibility Assessment report**
This report is produced at the Feasibility Phase of the DSDM process, it’s a short document that evaluates whether the application is technically and financially possible, and whether there are any major risks or constraints. For this product to be approved, it requires business and technical stakeholders to agree that the application is feasible with the available resources, time, and technology. It must be understood that there is nothing preventing progression to detailed planning. This is a Milestone product as it will be the same after this phase. In context of the scenario, it will be a document that evaluate whether the budget for the application will be enough from the start of development to the end of the development, it will also evaluate whether the application itself is technically viable (e.g. if a timeline of 6 months is enough, too much or too little).

iii. **Business Case**
This product is produced at the Foundations Phase of the DSDM process, it’s a justification for the application, outlining expected benefits, costs, risks, and return on investment. The business must approve the Business Case, confirming the value of the app, that’s how the Business case is approved and the development team must sign off on the Solution Architecture and Development Approach, to ensure technical feasibility. This is a Milestone product as it will not change after this phase. In context of the scenario, this will be a document given to Billy and Sarah to show them any profits, external costs or potential risks that they may face during the development of the application, it will show the overall value of the application.

iv. **Model/Prototype**
Developed at the Exploration Phase of the DSDM process, it is an early working model of features or user interfaces to give design direction, it has a detailed explanation of every individual component being developed. The Models/Prototypes are reviewed and approved by business users and stakeholders, ensuring alignment with expectations before full development continues. This is an Evolutionary product as it improves overtime. In context to the scenario this could be a version of the applications dashboard to display how icons were used over text and how users will easily navigate the application.

v. **The Evolving Solution**
This is developed the Engineering Phase of the DSDM process, it's a working version of the application that improves with each iteration, integrating all tested features. For this product to be approved the product must meet quality standards and pass functional, usability, and integration tests, a product increment is approved when it satisfies all "Must have" requirement. This is an Evolutionary product as it is improved overtime. In context of the scenario this will be the actual application that Billy and Sarah requested, with all working functionalities and all requirements implemented. (Lucid, 2025), (DSDM: The dynamic systems development method, 2024), (proj insights, 2023), (informIT, 2002), (Messenger, 2012)

TASK 4: Requirements/User Stories

Below are 5 user stories with an explanation of the difference between a User story and an Epic at the end.

i. As a business person with two devices of 2 different operating systems I need an application that is available on both operating systems so that I can access the application on both my android phone and iPad.

ii. As a regular user with no intention of selling any items on the application I want an application where I can simply just view the prices of items without making any payments so that I keep myself up-to-date with various product prices.

iii. As a business person with many passwords on different platforms, I want feature that allows me to login without having to manually type my credentials so that I can access the application even after forgetting my credentials.

iv. As a regular user that wants to use the application to buy and sell items, I want a feature that detects fake listings so that I don't get scammed.

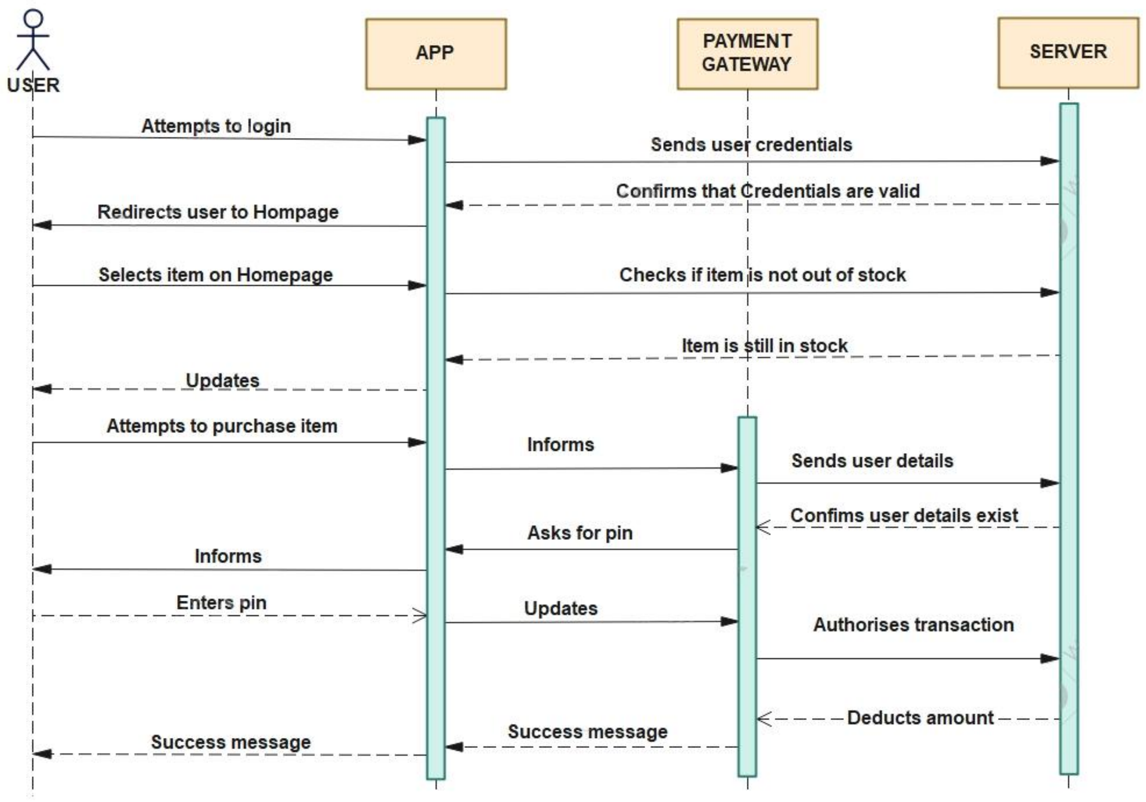
v. As a user that buys items very frequently, I want a feature that alerts me when a new item arrives on the marketplace so I can go check it out and decide whether to get it or not.

An Epic is a big, high-level body of work that represents a general goal, feature, or functionality in a system. An Epic is too big to complete in a single development cycle, so it is broken down into smaller, more manageable pieces, these pieces are called User Stories. Epics help structure and

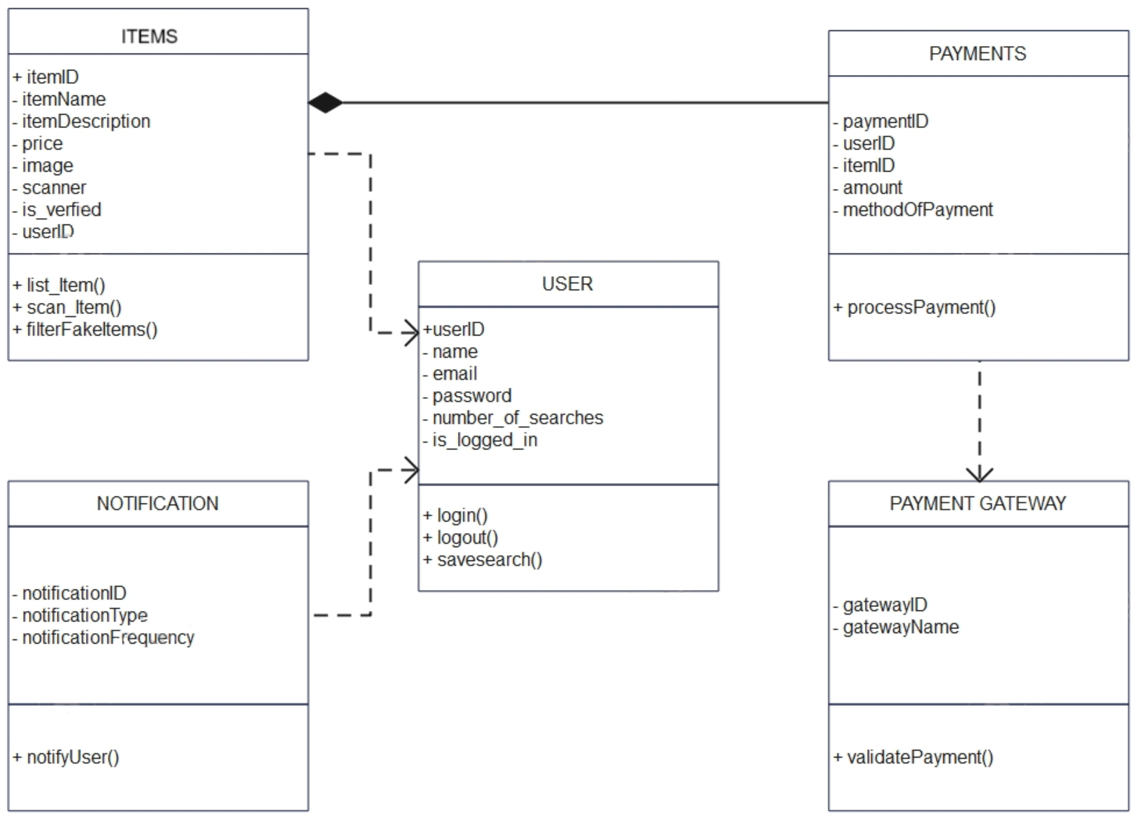
organize large ideas, making it easier for teams to plan and deliver in stages rather than trying to complete massive requirements all at once.
On the other hand, a User Story is a short, specific requirement from the view of the end-user. It describes a small, valuable piece of functionality that can be delivered within a sprint. A user story is written in this type of format: (As a [type of user], I want [goal] so that [reason]). This format ensures that the focus remains on the user's needs rather than just technical requirements.
In Agile project management, teams first create Epics to organize work around key functionalities. Once the team is ready to work on an Epic, the splits it into multiple User Stories that can each be estimated, prioritized, and assigned individually. For example, if the Epic is "Enable third-party logins," User Stories that relate to that epic might be, "Login with Google account", "Login with Facebook account" and " Securely handle user data during third-party login." Each user story can be completed in one sprint, whereas the full Epic will take multiple sprints.
Epics provide a high-level direction for the system's major goals, this makes sure the project remains focused on delivering bigger, meaningful features. User Stories make sure that the development work stays practical, achievable, and focused directly on user value. Both user stories and Epics work together to make large projects manageable, structured, and user centered.
In conclusion an Epic outlines the overall objective while a User Story defines the tasks needed to achieve a specific goal. An example of an Epic might be "Implement user authentication," while an example of User Stories under this Epic could be "User registration," "Login with email," and "Password recovery." These are more specific to the user's need, while an Epic is just a broader, general requirement. (What is DSDM in Agile? Guide to the 8 Principles of This Framework, 2021) (Rehkopf, 2025)

TASK 5: Modelling (Sequence Diagram, Class Diagram)

a) Sequence Diagram



b) Class Diagram



TASK 6: Prototyping

In software development, many teams often face uncertainty about tools, technologies, system designs, or whether certain ideas will even work in their projects. To reduce this uncertainty, developers use special methods called capability or technique prototyping methods. These help the team test out ideas before fully committing to building them into the final product.
Two common capability or technique prototype methods are:

- Architectural Spike
- Proof of Concept (PoC)

Both of these methods are used to explore unknowns and are also used to reduce risks, they do different things and are used at different stages of the development process.

Architectural Spike

This is a thin end to end mock-up of the pathway through a solution, it may be created in order to explore options and work out the best way forward, DSDM Atern calls these capability or technique prototypes (Messenger, The Dsdm Atern Student Workbook a Guide to the definitive agile framework, 2012).
Other sources describe it as a small and focused piece of experimental code or research created to evaluate technical approaches, especially around the architecture of the system.

Both these descriptions define an Architectural Spike as a product used to experiment techniques. An Architectural Spike is classified as a Disposable prototype.

Architectural Spikes are used for the following reasons:

- To explore solutions for a technical or architectural problem.
- To understand the complexity of a problem and estimate the effort that will be required.
- To help teams make informed decisions about how to build something.
- When the team is unsure how different parts of a system should be structured or connected.
- When there are multiple ways to build a system, and the best one isn't clear.
- When introducing new frameworks, tools, or design patterns into the project.

Characteristics of an Architectural Spike:

- Focused on technical architecture and design decisions.
- Usually not meant to be reused in the final product.
- Helps estimate the effort required for future work.
- Involves coding, but only enough to test a technical idea.
- It is Time-boxed.
- It is focused a specific technical question or challenge.

In context to the scenario, since Digital Disrupt is going to make the application on both iOS and Android, they might be uncertain on whether to use flutter or react native, an Architectural Spike would involve them creating some implementations on both frameworks to analyse factors like community support, ease of integration with existing systems and performance. The information they find after this will determine the technology digital disrupt will use

Proof of concept

A Proof of Concept is a prototype built to prove that an idea or solution is viable or possible. It focuses more on validating an idea or capability, not architecture, it is designed to test whether a certain idea or approach is feasible and aims to validate all assumptions made before allowing any resources to be committed.

Proof of concept is used for the following reasons:

- To show that a feature, technology, or business concept actually works.
- Sometimes used to demonstrate value to stakeholders or clients.
- to test if it's technically feasible.
- When you want to validate a new business feature before investing in full development.
- When trying out third-party tools, libraries, or APIs.

Characteristics of a Proof of concept:

- Focuses on testing feasibility of an idea.
- May include rough coding, mock-ups, or scripts.
- Often used to show stakeholders that an idea is possible.
- Can be reused partially in the final product, but not always.

- Aims to validate a specific concept or functionality.
- Focuses on critical aspects rather than full features.

In context to the scenario, Geekup is going to have an application with machine learning algorithm to detect fake Items a Proof of concept will be very important for this task.
The Proof of concept will uses a small dataset of fake and real items, the machine learning model will be trained and tested on the small set of datasets then depending on the progress made, the idea of having a machine learning model to detect fake items will be either labelled feasible or impossible.

Below are key differences of Architectural Spike and Proof of concept:

- Architectural Spike produces insights, documentation or prototypes while Proof of concept produces working prototypes to demonstrate feasibility.
- Architectural Spike is short and time boxed while Proof of concept depends on complexity of the product.
- Architectural Spike explores technical challenges and focuses on the architecture of the product while Proof of concept validates feasibility of a concept or a technology.

These two tools are pretty similar but they differ in different ways including when to use them. Relating to the scenario, Geekup will use an Architectural Spike When facing uncertainty about which framework to use when integrating a new operating system, they will also use it to analyse the pros and cons of integrating new technologies like adding a machine learning model to detect fake items, lastly, they could use it before estimating effort for complex features like the functionality allowing user to save searches.
Proof of Concept would be used when Digital Disrupt want Billy and Sarah to validate a certain idea or approach and before committing to full-scale development.

Both Architectural Spikes and Proofs of Concept are valuable tools in Agile development, serving distinct purposes. Architectural Spikes help teams navigate technical uncertainties and make informed architectural decisions, while Proofs of Concept validate the feasibility of ideas or technologies before significant investment. Understanding when and how to use each can lead to more efficient and successful mobile app development projects.

TASK 7: Retrospective

a) Timeboxing is an Agile project management technique where an activity is allocated a fixed amount of time. The activity is meant to be completed within the time allocated; the team either finishes within the allocated time or stops and reviews progress. The main focus is completing the most valuable work possible within a given timeframe, instead of allowing tasks to expand endlessly. Timeboxing encourages productivity, decision-making, prioritization of important features, and frequent delivery of usable work. (Timeboxing, Chapter 13:, 2025), (Messenger, The

Dsdm Atern Student Workbook a Guide to the definitive agile framework, 2012)

In other words, you set a strict deadline, and whether the work is perfect or not, you move forward after that time.
For Geekup, Timeboxing will be used to break the overall timeline into specific tasks for major elements like requirements gathering, UI/UX design, backend development, frontend development, testing, and deployment.
The team at Digital Disrupt will agree on whether or not to extend deadlines for each phase. If something cannot be finished perfectly within its timebox, they will deliver the best workable version or prioritize critical parts.

Below is how Digital Disrupt would structure the tasks using Timeboxing:

i. **Requirements Gathering and Initial Planning**
Estimate: 2 weeks.
Focus mostly on core features like login, search, notifications, and responsiveness.
This prevents wasting months gathering unnecessary features and allows time to interview stakeholders and draft the product backlog.

ii. **UI/UX Design**
Estimate: 3 weeks.
1 week for wireframes and user flow diagrams.
2 weeks for detailed mock-ups of major pages like Home, Search, Items, and Profile.
Focus is on simplicity and usability, not endless beautification.

iii. **Backend Architecture and Database Setup**
Estimate: 4 weeks.
Build server infrastructure, database models, and APIs for users, items, payments, and notifications.
Start with necessary features and, if time allows, add complex features like machine learning later.

iv. **Frontend Development**
Estimate: 7 weeks.
Implement major pages like Login/Register, Search, Post Item, Purchase Item, and Profile Management.
Strictly focus on functionality, not advanced animations.

v. **Machine Learning Integration for Fake Item Detection**
Estimate: 4 weeks.
2 weeks for dataset collection and model training.
2 weeks for integration with item listings.
Focus on delivering a 60–70% accuracy model to avoid wasting time chasing 100% accuracy.

vi. **Testing and Quality Assurance**
Estimate: 5 weeks.

2 weeks for internal quality assurance.
2 weeks for Billy and Sarah to test the app and give feedback.
1 week to fix critical issues before release.

vii. **Marketing Preparations and Launch Readiness**
Estimate: 2 weeks.
Prepare app store listings, documentation, screenshots, and customer support setup to avoid last-minute launch chaos.

viii. **Buffer and Minor Adjustments**
Estimate: 2 weeks.
Reserved for unexpected delays or final polishing. Focus will be on stabilization, not adding "nice-to-have" features.

In conclusion, Timeboxing is effective because it:

i. **Focuses on Business Value**: Forces the team to prioritize important features over unnecessary extras.

ii. **Avoids Perfectionism**: Ensures delivery of a working app even if not everything is perfect.

iii. **Has Predictable Progress**: Billy and Sarah will track what will be completed and when.

iv. **Encourages Team Motivation**: Achievable goals prevent exhaustion compared to one massive, never-ending project.

Using Timeboxing in the 6-month period will ensure speed, focus, and delivery of working software instead of getting stuck in endless planning and perfectionism. (Indeed Editorial Team, 2025)

b) For a project like GeekUp's using DSDM is the most suitable choice. DSDM is known for its disciplined approach and business-driven development, however, other agile approaches, SCRUM particularly, could offer a more effective fit for GeekUp's unique circumstances.
Below are 5 reasons why SCRUM would be a good approach in this scenario:

i. **Team Size and Resource Constraints**

One of the major challenges for Geekup is it has limited staff. With only two stakeholders (Billy and Sarah) and no in-house developers, the project relies entirely on Digital Disrupt an outsourced software house. DSDM assumes there is a dedicated team available with clearly defined roles like business ambassador, business analyst, technical coordinator, and solution developers/testers. This is difficult to maintain in a setup where the internal team is small and external development is contracted. Scrum is designed to be lightweight and adaptable to small, cross-functional teams. It works well with limited resources and allows role flexibility which is something crucial for startups like Geekup.

ii. **Customer Involvement and Rapid Feedback**

DSDM strongly emphasizes continuous user involvement and well-defined user roles throughout the lifecycle, however, in reality, startup owners like Billy and Sarah may not have time or technical background to qualify for roles like business visionary or business ambassador full-time. Scrum allows changing availability of stakeholders more easily, focusing on incremental delivery and quick feedback through short development cycles called sprints. Scrum allows stakeholders to provide feedback at the end of each sprint and during the Sprint Review even if they can’t participate in the full process.

iii. **Timeboxing and Flexibility**

DSDM priorities strict timeboxing and sets fixed timelines for delivery of each increment, this ensures predictability but it may not ensure the flexible and often unpredictable nature of a startup company like Geekup. Geekup is operating under a six-month deadline and a fixed budget of £25,000, but their app includes advanced features like machine learning for counterfeit detection and integration with third-party login services. These components are very exploratory and require a more iterative and flexible approach, such as Scrum where progress is tracked continuously and adjustments are made based on feedback and learning.

iv. **Innovation and Evolving Requirements**

Startup companies like Geekup often change their product strategy based on user feedback and competitor activity. DSDM locks in high-level requirements early in the process and discourages major changes after the Foundations phase. While it does allow for requirement changes at lower levels, Scrum allows changing requirements and is built around change. Each sprint offers a chance to refine functionalities and adjust content based on actual market feedback, which is important when innovating in a competitive space like online trading platforms.

v. **Focus on Working Software**

Scrum’s primary focus is on delivering working software at the end of each sprint, making it ideal for startups that want to test features quickly and incrementally. DSDM places a strong emphasis on documentation and structured governance, which can slow down delivery. Geekup can benefit more from Scrum’s rapid prototyping and early user feedback, especially with features like the search functionality, real-time notifications, and the machine learning algorithm, this would require validation from users for better improvements.

While DSDM offers a good structure, it is rigid, has formal processes, and timeboxing may not align well with Geekup’s small team, flexible vision, and need for rapid iteration. On the other hand, Scrum is lightweight, flexible, and has a feedback-driven nature which makes it a better choice for this project. (PoC vs prototype vs MVP in mobile app development, 2024), (Arceo, 2023)

REFERENCES

Agile Business. (2025, febuary 17). *What is DSDM?* Retrieved from Agile Business Consortium: https://www.agilebusiness.org/business-agility/what-is-dsdm.html

Agile Business Consortium. (2025, Febuary 08). *Chapter 15: Requirements and user stories*. Retrieved from Agile Business Consortium: https://www.agilebusiness.org/dsdm-project-framework/requirements-and-user-stories.html

Arceo, S. (2023, august 1). *Spike In Scrum: Definition, Benefits, And How To Use Them*. Retrieved from Mambo.IO: https://mambo.io/blog/spike-in-scrum?

DSDM: Definition, Benefits, Principles and Practices. (2025, april 10). Retrieved from indeed: https://www.indeed.com/career-advice/career-development/dsdm

DSDM: The dynamic systems development method. (2024, May 16). Retrieved from LogRocket: https://blog.logrocket.com/product-management/dsdm-dynamic-systems-development-method/

Dynamic Systems Development Method (DSDM). (2025). Retrieved from Product Plan: https://www.productplan.com/glossary/dynamic-systems-development-method/

Dynamic Systems Development Method (DSDM): Driving Agile Project Success. (2023, MAY 06). Retrieved from The Agile Handbook: https://www.theagilehandbook.com/blog/dynamic-systems-development-method/

Dynamic Systems Development Method (DSDM): Driving Agile Project Success. (2023, May 06). Retrieved from The Agile Handbook: https://www.theagilehandbook.com/blog/dynamic-systems-development-method/

Indeed Editorial Team. (2025, April 10). *DSDM: Definition, Benefits, Principles and Practices*. Retrieved from indeed: https://www.indeed.com/career-advice/career-development/dsdm

informIT. (2002, march 22). *A Human Being Is Not Equivalent to a Tool: Working with the Dynamic Systems Development Method*. Retrieved from informIT: https://www.informit.com/articles/article.aspx

Janse, B. (2025, April 10). *Dynamic Systems Development Method (DSDM) explained*. Retrieved from toolshoro: https://www.toolshero.com/information-technology/dynamic-systems-development-method/

Lucid. (2025, march 01). *Dynamic Systems Development Method (DSDM)*. Retrieved from aairfocus: https://airfocus.com/glossary/what-is-dynamic-systems-development-method/

Lyoshin, V. (2025, january 12). *Dynamic Systems Development Method (DSDM)*. Retrieved from Vit Loshin: https://vitlyoshin.com/blog/dynamic-systems-development-method/

Messenger, S. (2012). *The Dsdm Atern Student Workbook a Guide to the definitive agile framework.* Chechire: Galatea Training Services Limited.

Messenger, S. (2012). *The DSDM Atern student workbook a Guide to the definitive agile framework.* Chechire: Galatea Training services Limited.

PoC vs prototype vs MVP in mobile app development. (2024, august 1). Retrieved from https://tapptitude.com/blog/poc-vs-prototype-vs-mvp-app-development

proj insights. (2023, May 11). *What is DYNAMIC SYSTEMS DEVELOPMENT METHOD and its 8 Principles*. Retrieved from projinsights: https://www.projinsights.com/what-is-dynamic-systems-development-method-and-its-8-principles/

Rehkopf, M. (2025, April 2). *Stories, epics, and initiatives*. Retrieved from ATLASSAN: https://www.atlassian.com/agile/project-management/epics-stories-themes

Richards, K. (2023, febuary 20). *DSDM (Dynamic Systems Development Method)*. Retrieved from agilekrc: https://agilekrc.com/agile/dsdm

Scholz, R. (2025, Febuary 24). *Why DSDM Is the Optimal Choice in Times of Limited Investment*. Retrieved from netguru: https://www.netguru.com/blog/why-dsdm-is-the-optimal-choice

Timeboxing, Chapter 13:. (2025). Retrieved from Agile Business Consortium: https://www.agilebusiness.org/dsdm-project-framework/13-timeboxing.html?utm_source=chatgpt.com

Venema, M. (2024, April 26). *Dynamic System Development Method (DSDM):*. Retrieved from nimble work: https://www.nimblework.com/agile/dynamic-system-development-method-dsdm/

What is DSDM in Agile? Guide to the 8 Principles of This Framework. (2021, April 15). Retrieved from appviser: https://www.appvizer.com/magazine/operations/project-management/dsdm

Zartis Team. (2025, january 2). *Different Agile Methodologies with Pros & Cons*. Retrieved from Zartis: https://www.zartis.com/different-agile-methodologies-with-pros-cons/