Sprint 02 Planning
Group Task 16P

Team Name: Team Grotle
Tutorial Class: Tue 12:30 EN310
Tutor: Naveed Ali
Team members:

Group members	Student ID
Hamish	103607352
Dilni	103616345
Kamar	103607585
Melanie	103489466
Justin	102589705
Tevy	103139978
Cormac	102581060

Backlog items

Sprint 2 updated backlog items

No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 | 2)
F1	Add a new member (remake in PHP)	-	10	Sprint 2
F2	Deactivate member (remake in PHP)	F1	8	Sprint 2
F3	Search for member (remake in PHP)	F1	2	Sprint 2
F4	Edit existing member (remake in PHP)	F1	2	Sprint 2
F5	Add a sales record	F1	10	Sprint 2
F6	Record in the spreadsheet of purchase logs and product levels. Downloadable in CSV format	F1, F5	9	Sprint 2
F7	Search for sales record	F5	2	Sprint 2
F8	Update sales record	F5	7	Sprint 2
F9	Remove Sales record from view	F5	1	Sprint 2
F10	Produce a daily sales report on popular items (what was sold the most) Downloadable in CSV format	F5	8	Sprint 2
F11	Produce a weekly sales report on popular items (what was sold the most) Downloadable in CSV format	F5	8	Sprint 2
F12	Produce a monthly sales report on popular items (what was sold the most) Downloadable in CSV format	F5	8	Sprint 2

Prioritising criteria

Factor (Most - Least Important)
1. Feature Dependency - Feature dependencies for sprint 02 are considered one of the most important criteria. Featured in the web application ultimately depends on the completion of another. As functions build on top of each other. Such as the backlog item F6 which is dependent on items F1 and F5. Furthermore, this is demonstrated in the purchase logs section of the product level and purchase logs report. The adding member's function is needed to record the purchase logs. As which member

purchased what item is needed. Hence without the member being registered this function wouldn't be functional, hence needing F1 to be completed before F6. In addition, F6 will also need to have a way to see the sales record or F5 as the product level is also dependent on its completion.

- It is also noted that F1 - F4 Items are completed sprint 01 items, however, are being remastered from java to PHP. As well as not completing the items from sprint 01 which are F5 and F6.Hence due to this, these are items are even more important to have completed as a lot of sprint 2 (F7 - F12) items are dependent on sprint 01 items.

2. Risk Management/ risk Involvement
- Risk management criteria would be considered important criteria because if we do not consider the things that could go wrong and affect the project we will not have a way to prepare for and deal with that problem. For instance, if we had planned to allocate F5- F8 2 hours to complete but we end up needing more time, the effect is that tasks F9-12 get delayed or not much time can be dedicated to them. Another risk might be skill or technology related. For tasks F10-F12 we need to implement a way to generate a report and download it, this may be something that requires research if we have not done something like this before which will increase the amount of time and effort needed, the risk being that it may prove to be a challenging task.

3. Development Effort
- Development is the third-highest priority, as it determines the amount of time and resources needed to complete backlog items. Which is a factor that influences the decision to prioritise certain items over another. Evidently seen in the first prioritisation of F1 Add members to be transformed into PHP from java. This is selected first as it would be the easiest to implement as the logic is already there for the function and ultimately it's the changing of language that needs to be altered. This is also similar to the F6 prioritising second as a lot of the resources are already there and the function itself doesn't take long to implement and is fairly simple. Therefore, it is prioritised first before other items.

4. Date/Timeline
- Data/timeline is an important factor as we have seen from our first sprint. It is listed as 4th most important because the data/timeline is covered under risk management. When taking into account the risks that could affect a task's completion this will ultimately result in the date/timeline being changed to adjust for the possible occurrence of risks. However, it was not considered as much as other factors as the development team wanted to concentrate on more completion of tasks and functions and not stress about the completion dates for section backlog items. But the completion date of the sprint is still very much considered.

Backlog item prioritising

Pos.	No.	Item	Rationale
1	F1	Add a new member (remake in PHP)	This item was completed in the last sprint but needs to be remade in PHP since our team has decided to develop the project as a PHP application. This item takes the highest priority since backlog item F5 depends on it for development. According to the criteria, feature dependency is the first criteria that need to be fulfilled.
2	F5	Add a sales record	This item was meant to be done in the last sprint and its implementation is crucial for most Sprint 2 backlog items. Hence according to the first highest criteria of the prioritising criteria, feature dependency due to the item having a large portion of sprint 02 items dependent on it will be prioritised first. In addition to this Due to it being a sprint 1 item, there is considerable risk in its completion. As if it has to be completed this sprint or this main feature will not be implemented affecting other dependent items. Hence fulfilling the second-highest criteria of the prioritising criteria
3	F6	Record in the spreadsheet of purchase logs and product levels. Downloadable in CSV format	This item was meant to be done in Sprint 1. However, due to the lack of time, it was not able to be completed and hence transferred to Sprint 2. It is also one of the harder items to develop, as our team is not very experienced in report generation. Thus, it requires more development effort than the tasks below and is more prone to bugs and errors due to its complexity. So this item is related to development effort criteria which is the third-highest priority.
4	F10	Produce a daily sales report on popular items (what was sold the most) Downloadable in CSV format	Backlog item F6, the most comparable task, takes a considerable amount of effort to develop. Thus, these items will most likely take a similar amount of development effort. Like backlog item F6, they are also prone to bugs and errors due to their complexity. These items are in the development effort criteria and also risk involvement criteria, which make them also a high priority.
5	F11	Produce a weekly sales report on popular items (what was sold the most)	

		Downloadable in CSV format	
6	F12	Produce a monthly sales report on popular items (what was sold the most) Downloadable in CSV format	
7	F2	Deactivate member (remake in PHP)	These items were completed in the last sprint but need to be remade in PHP since our team has decided to develop the project as a PHP application. Seeing as these items are some of the simplest to make, as they just need to be remade in a similar language, they're low priority. These items do not need much development effort, plus there are not many risks involved if we could not remake them.
8	F3	Search for member (remake in PHP)	
9	F4	Edit existing member (remake in PHP)	
10	F8	Update sales record	Seeing as these items are the simplest to make which is not much development effort. As they're just additions to backlog item F5, they're the lowest priority. This also correlates to the third-lowest of the prioritising criteria. Their inclusion also isn't as important as F5 or F10-12 due to the functions these tasks perform being achievable through making a physical note. Therefore there is not much risk to not having these items completed. According to the second-highest prioritising criteria.
11	F7	Search for sales record	
12	F9	Remove Sales record from view	

Velocity planning

Sprint 1 planned and actual working hours	Sprint 02 Planned working
Total hours = 112 Remaining hours = 12 Completed hours = 100 10% → 112 = 11.2 Above = 123.2hrs Below = 100.8 hrs	10% → 100 = 10 Above = 110hrs **Below = 90hrs**

During the completion of sprint one, the development team had a velocity of 100 hrs during the working week. Which is 10% or 11.2 hours below the planned working hours. Which is

100.8 hrs. Additionally, the team didn't complete 2 backlog items as well. Evident in the remaining 12 hours of working time left. Furthermore, due to this for sprint 02 the development team decided to work 10% below or 10 hours less than the actual velocity which would be 90 hrs. This will enable us to complete good quality work and not have to rush to finish sprint 2. Whereas if we went 10 % or 10 hours above. This will lead to bad quality work which may have to be re-tested and fixed, therefore leading to an unfinished product.

Effort estimation edit and updates

To calculate the estimated time of completion for each task, size comparison and Analogy Comparison was used to come to the estimation of sprint 2 backlog items. Using actual times collected from sprint 1 backlog items that were similar to sprint 2 backlog items. Hence, obtaining an accurate estimation for sprint 2 task completion times is seen below.

Backlog items and effort estimation	Comparable items and time effort	Method	Explanation
F1: Add a new member (remake in PHP) [1hour]	**Sprint 01** F1 - Add a new member [2hours]	Size Comparison	F1 sprint 1 took 2 hours to complete. Both backlogs are in a very similar size, but F1 in sprint 2 will be **0.5 effort** since it is just converting to PHP (without implementing a new html page and SQL). So remaking "Add a new member" to PHP would take around **1hour.**
F2 : Deactivate member (remake in PHP) [1hour]	**Sprint 01** F2 - Deactivate members [2hours]	Size Comparison	F2 sprint 1 took 2 hour to complete. Both backlogs are in a very similar size, but F2 in sprint 2 will be **0.5 effort** since it is just converting to PHP. So remaking "Deactivate members" to PHP would take around 1**hour**.
F3: Search for member (remake in PHP) [1hour]	**Sprint 01** F3 - Search for member [2hours]	Size Comparison	F3 sprint 1 took 4 hours to complete. Both backlogs are in a very similar size, but F3 in sprint 2 will be **0.25 effort** since it is just converting to PHP without a new implement HTML page and SQL. So remaking "Search for member" to PHP would take around 1**hour**.
F4: Edit existing members (remake in PHP) [1hour]	**Sprint 01** F4 - Edit existing members	Size Comparison	F4 sprint 1 took 2 hours to complete. Both backlogs are in a very similar size, but F4 in sprint 2 will be **0.5 effort** since it is just

	[2hours]		converting to PHP without implementing a new html page and SQL. So remaking "Edit existing members " to PHP would take around 1**hour**.
F5: Add sales record [2 hours]	**Sprint 01** F1 - Add new member [2 hours]	Size Comparison	As backlog items, F5 is similar to items F1 from sprint 1 which took 2 hours to complete. Hence it would be assumed that due to F5 similarities to F1, F5 would have a completion time of **approx 2 hours** as well. This can be broken down into - Implement Html [30 minutes] - Implement SQL [30 minutes] - Implement Php [1 hour]
F6: Record in the spreadsheet of purchase logs and product levels. Downloadable in CSV format [4 hours]	A User- entering web application where details are downloadable in CSV file format. [4 hours]	Analogy Comparison	Previously there was a similar project completed by the development team which had a function where users' details were printed out into a CSV report. This task took 4 hours to complete. Hence due to its similarities to this sprint 2 F6 task. The estimation by analogy for the F6 would also be **4 hours approx**. This can be broken down into - Generate report functionality [3 hours] - Researching how to complete download report feature [30 minutes] - Implementing download report [30 minutes]
F7: Search for sales record [4 hours]	**Sprint 01** F3 - search for member [4 hours]	Analogy Comparison	Estimation of the time taken to complete F7 is obtained by comparison with backlog item F3 from sprint 1. Using the estimation analogy method. Moreover, F7 and F3 are both search functions. Alternatively, searching for different data. Hence it would be assumed that the time taken to complete F7 is around **4 hours approx**.

			This can be broken down into - Create Html page [1 hour] - Come up with SQL query [30 minutes] - Develop and implement PHP page [2 hours and 30 minutes]
F8:Update sales record [2 hours]	**Sprint 01** F4 - Edit existing member [2 hours]	Analogy Comparison	To estimate the time allocation for task F8. It is compared to task F4 from sprint 1 by the estimation method analogy comparison. Due to their similar functionality of editing / updating the data for a given variable. Furthermore, F8 by its similarity to F4 should take **2 hours appro**x to complete. This can be broken down into - Create html page [1 hour] - Develop and implement SQL query [30 minutes] - Implement PHP page [1 hour 30 minutes]
F9: Remove Sales record from view [2 hours]	**Sprint 01** F2 - Deactivate member [2 hours]	Analogy Comparison	To estimate the approximate time needed to complete F9, it is compared accordingly to F2 from sprint 1. Implementing the analogy comparison method. As a result, both backlog items have very similar functions of removing saved data from the view of the user. Hence it is estimated that F9 will take **3 hours approx**, similarly to F2. This can be broken down into: - Create and code Html page [1 hour] - Develop and implement SQL query [30 minutes] - Implement and develop PHP page [1 hour 30 minutes]
F10: Produce a daily sales report on popular items (what was sold the most) Downloadable in CSV format [4 hours]	A User-entering web application where details are downloadable in CSV file formats. [4	Analogy Comparison	Backlogs item F10 has a similar download function of a similar software project. Using estimation by analogy method, it is estimated that F10 will take **4 hours approx** to be completed. This can be broken down into

	hours]		- Generate report functionality [3 hours] - Researching how to complete download report feature [30 minutes] - Implementing download report [30 minutes]
F11: Produce a weekly sales report on popular items (what was sold the most) Downloadable in CSV format [4 hours]	A User-entering web application where details are downloadable in CSV file format [4 hours]	Analogy Comparison	F11 has a similarity to the download function of a similar software project. Therefore using the analogy comparison method, it is estimated that F11 will take **4 hours** to be completed. This can be broken down into: Generate report functionality [3 hours] Implementing download report [1 hour]
Set Up	Set up work done in **Sprint 01**	Analogy Comparison	Before writing code in Sprint 1, 4 hours (that were not counted towards our work hours) were used to set up the: - Netbeans IDE [2 hours] - database of the application [1 hour] - basic navigation pages [1 hour] We've decided to add this backlog item to Sprint 2 so that we can allocate sufficient time for this setup work, instead of rushing to set things up in our off-time. Using the analogy comparison method, it is estimated that this backlog item will take **2 hours** to be completed. This can be broken down into: - Set the database up [1 hour] - Create basic navigation pages [1 hour] Note: There is no time allocated for setting up the NetBeans IDE since we will no longer be using it in Sprint 2.
Implement CSS	A User-entering web	Size comparison	In sprint 2 implementations of the CSS are estimated using the size

| | application where details are downloadable in CSV file format.

CSS implementation [10 hours] | | comparison method. As both tasks had an equal number of **5 developers** working on the CSS implementation. In the development of this previous user entering a web application, it took **10 hours** of work time to complete the CSS implementation in the header file. Which was split between **5 developers**, hence **2 hours** each for working time. Therefore, using the size comparison method, sprint 2 CSS implementation would similarly take **10 hours approx.** That will be split between **5 developers** as **2 hours each approx.** |