Sprint Practice Example

Exercise 1 - identify product goal, user personas and their needs and features

Using Roman’s Product vision board template ().

**Goal: Double our business by growing our reach**

| **Persona** | **Needs** | **Product (High level deliverables)** |
| --- | --- | --- |
| Students | Make ordering fun | Build your own pizza |
|  | Tell a friend | Vouchers |
|  | Pay in cash | Pay on delivery |
| Late-night workers | Make it easy to order for a team | Multiple items |
|  | Easy to pay | Pay online |
|  | When to go out to meet the driver | Delivery alerts |
| Busy families | Easy to order for a family and friends | Combos and suggestions |
|  | Save money | Promotional discounts |
|  | Keep kids aware of how far away it is | Driver tracker |
| Pizza shop staff | Easy to update with new choices | Menu administration |

Exercise 2 - Create Product Backlog

2.3 Product Backlog

| **User story ID** | **User story** | **Priority** | **Acceptance Criteria** | **Estimate** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
| U001 | As a *late-night worker* I can *order multiple items* so that *it is easy to order for the team*. | 1 | Can list all available pizzas on the menu.  Can add multiple items to the basket.  Cannot confirm order if not logged on. Response time must be under 2 seconds. |  |
| U002 | As a *late-night worker* I can *pay online* so that *it is easy to pay.* | 2 |  |  |
| U003 | As a *late-night worker* I can *get delivery alerts* so that *i know when to go outside to meet the driver.* | 3 |  |  |
| U004 | As a *student* I can *build my own pizza* so that *ordering is fun.* | 4 |  |  |
| U005 | As a *student* I can *pay on delivery* so that *I can pay in cash.* | 5 |  |  |
| U006 | As a *pizza chef* I can *see the next pizza order* so that *I make pizzas in the right sequence.* | 6 |  |  |
|  |  |  |  |  |

*User story: As a [type of user] I can [do an action] so that [I get a result]*

*Acceptance Criteria :* Everything that is important to test that the story is working as required. These are statements that define what the product must do and not-do. It specifies both functional (eg. how the functionality enables the user to achieve a goal)

and non-functional requirements (eg. access and authentication of the user). Each criteria should have a pass or fail condition.