**Document 02 – Sprint 1 Plan**

This document is contained in your GitHub repository in a folder named *docs*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Group | 2 | | Group Member Names |  |
|  | |  |  | | --- | --- | | 1. | Kevin Patel | | 2. | Tarun Patel | | 5. | Dongwoon Jeong | | |  |  | | --- | --- | | 3. | Utsav Patel | | 4. | Ho yong Lee | |
|  |  |  |

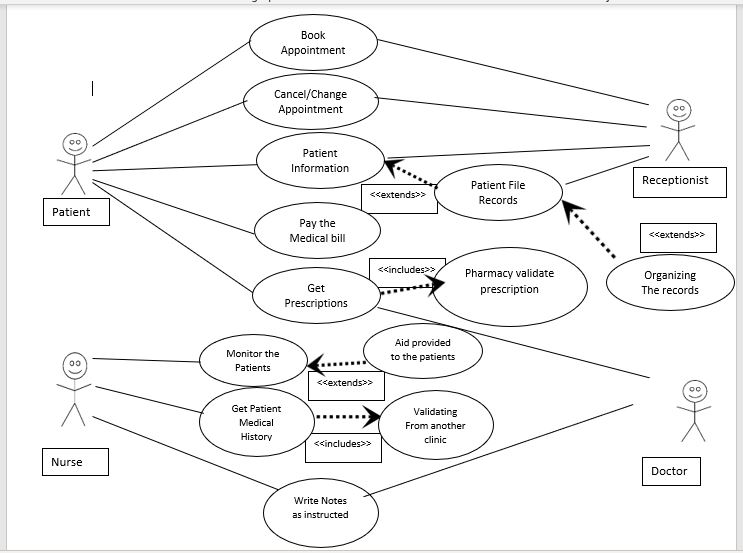
1. **Actors**

|  |
| --- |
| **Instructions**  Sections 1,2, and 3 are all related.  You will develop at least 10 use cases (more is fine). These should be the most important ones. In a later section, you will rank these on priority.  You should brain storm who the actors are and what they want to accomplish. For example: (1) a customer wants to book a flight, (2) a customer wants to book a flight with companions. (3) the airline wants to know how much revenue is generated from a flight.  **Deliverable**  Provide a numbered list of the Actors you have modelled and a brief description of each. |

|  |  |  |
| --- | --- | --- |
|  | **Actor** | **Description** |
| 1. | Patient | Patient can go see the doctor and see what is wrong with him |
| 2. | Receptionist | Receptionist at the doctor can make appointments and answer calls |
| 3. | Nurse | When you go to the Clinic, nurse can check your weight, Bp, and many more. |
| 4. | Doctor | The doctor can check to see what is wrong with and give you medicine for the illness. |

1. **Use Case Diagram(s)**

|  |
| --- |
| **Deliverable**  Provide a neat, legible, properly sized UML use case diagram(s). Accompany these with any discussion that is necessary. It is acceptable to move the diagram(s) to the next page if needed for it to be displayed optimally. |



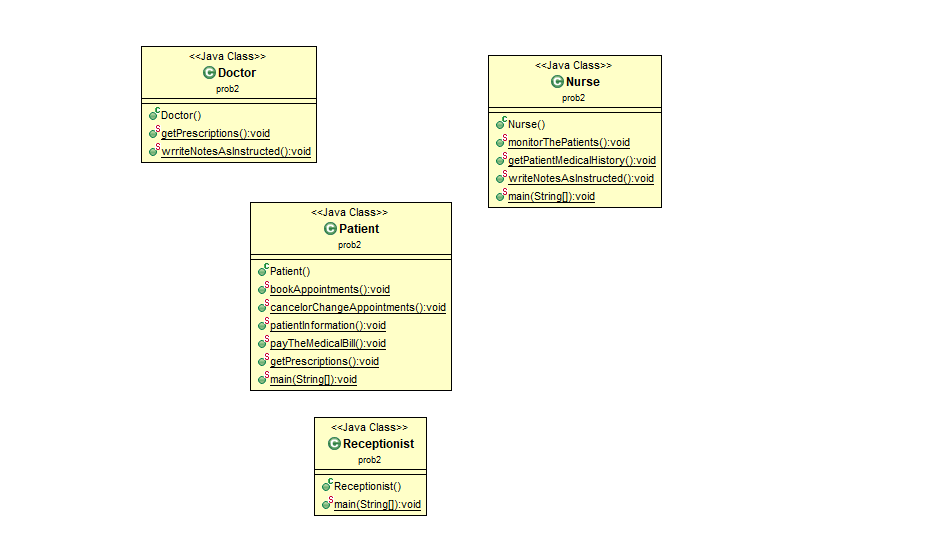
1. **Use Case Descriptions**

|  |
| --- |
| **Deliverable**  Provide a numbered list of use cases ordered by their priority and a brief description of each. |

|  |  |  |
| --- | --- | --- |
| **Priority** | **Use Case Title** | **Description** |
| 1. | Can book appointments | Patients can book appointments |
| 2. | Can cancel or change appointments | Patients can cancel or change the appointments |
| 3. | Paybill | Patients can pay the bill |
| 4. | Get prescription | Patients can get the prescription from the doctor |
| 5. | File Records | Receptionist needs to file records for the patients |
| 6 | Organize Records | Receptionist organize the records for the patients |
| 7 | Claim Insurance | Receptionist needs to claim the Insurance for the clinic |
| 8. | Monitor Patients | Nurse needs to monitor patients |
| 9. | Get patient medical history | Nurse needs to get the patients’ medical history for another clinic if any |
| 10. | Aid patients | Nurse needs to write down how they aid the patients |
| 11. | Write doctors notes | Nurse needs to write down any notes that the doctor gives her |
| 12. | Clock-in/clockout hours | Everyone needs to clock-in when they come in and clock-out when they leave |
| 13. | Manage Staff | Doctor need to see who he wants to hair and where he needs to put them. |
| 14 | Advise nurse for treatment | Doctor can tell the nurse what type of medicine they need to give to the patients |
| 15. | Write prescriptions to patients | Doctors need to send prescriptions to the pharmacy. |

1. **Class Diagram**

|  |
| --- |
| **Deliverable**  Consider the first 5 (more is fine) or so use cases and develop a neat, legible, properly sized UML class diagram(s) showing your initial design. Classes should show state and behavior. It is not important to capture every detail. Every time you look at your document, you will see new things you didn’t think of before: a required instance variable, a parameter for a method, a missing method, a method in the wrong class, a new class, etc. The important point is to establish a starting point for the design. |

****