**<System Name> Project Charter**

<*The project charter represents the first blueprint of the system. It is a statement of intent by your client of their desire to develop a software solution. The project charter is often updated as the project progresses and the requirements become clearer.>*

*The project charter is a contract with your user. When consulting the project charter is the basis for measuring your success in delivering the system the client wants.*>

1. **Systems Objective**  
   *<A short descriptive statement about the purpose of the project form the user’s perspective. It should be short, typically no more than 4 or 5 sentences.>*
2. **Project Roles/Responsibilities**  
   *<These represent the roles and names of each member of the team and the responsibilities that each has. Include users who will assist with the project.>*
3. **Business Benefits**

**Ben’s Test Change**  
*<This is a list of benefits that the solution will deliver. The benefits are stated from a business/user perspective. An example might be “the system will increase Internet sales by 20%”.>*

1. **Requirements**
   1. **Functional Requirements** *<Functional requirements describe the tasks that the system must perform to satisfy the clients' requirements. Apply event analysis to fill in the event table below.>*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Event** | **Source Actor** | **Use Case Name** | **System Response** | **Destination Actor** |
|  |  |  | **Source Actor + Verb + Object** |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |

* 1. **Non-functional Requirements** *<Non-functional requirements describe quality characteristics required for the new system. Non-functional requirements are not the same as functional requirements, which concern what the system must do. Typical non-functional requirements include ease of use, maximum response time, frequency of backup and system availability. An example is “All transactions must provide less than 2 seconds response time”. Refer to the Wikipedia article* [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirement) *for an exhaustive list. You are not expected to use all of the items listed in the article, just a subset that you think is relevant for the system that is being developed. Document non-functional requirements in a bulleted list.>*
* non-functional requirement
* non-functional requirement
* non-functional requirement
* non-functional requirement
* non-functional requirement

1. **Project Critical Success Factors**  
   *<These represent the factors that are required to make the project a success and factors which indicate that the project has been a success. An example might be “The technical staffing level and technical staff time commitment must not drop".>*
2. **Preliminary Technical Architecture**  
   *<The preliminary technical architecture represents technologies that are under consideration for the application.>*
   1. **Development Environment**
   2. **Server Production Environment**
   3. **Client Production Environment**