1. **overview :** (introduction)

This section display the main architectural project. Firstly it’s presented a high level presentation about our S2B and their interaction. Afterwards the document shows in a more deeply way different kinds of view about the system. Eventually it describes other various features about design and architecture decision

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **high lvl components and interactions**(functional view : Definiscono l’allocazione funzionale dei diversi componenti. Una vista funzionale definisce quindi i componenti e i connettori e controlla se gli assegnamenti alle funzioni corrispondenti sono completi.-->Durante la definizione della vista funzionale, si individuano i componenti e i connettori e si verifica che tutte le funzionalità richieste vengano assegnate ai componenti così individuati.)

This section aims to provide, through the diagram below, a general overview about system’s architecture and main interactions between each component.

(Diagram)

The architectural style appointed is a client-server application with a multi-tier style: a client tier, a tier for the business logic and a tier for the data. All main communications between each component and, or, tier happen in an asynchronous way.

**Client Tier:** this layer represents the different methods of access to the application: via one of the main common browser(e.g.: firefox, chrome, etc…), or through the mobile application, available for main mobile OSs. This logic level has the duty of representing information to the users and interacting with them( this can be seen as the View part of the MVC pattern). The Web Client is linked to a Web tier

This layer represents the different methods of access to the application: via one of the main common browsers (e.g.: Firefox, Chrome, etc…), or through the mobile application, available for main mobile OSs. This logic level has the duty of representing information to the users and interacting with them (this can be seen as the View part of the MVC pattern). The Web Client communicates with the business logic only through a Web tier.

**Web Tier:**  this layer contains a Web Server that functions as middleware between the Web Client and the Business Logic Application.

**Business Tier:**  This layer represents the system's core in fact all the application logic is inside this tier. It has three main components: one represents the Manager of all the rides, another one the Queue Manager and in the end a component which handles all the connections with the front end side, called Connection Handler. The Business tier can also access to the data in the Data tier.

**Data Tier:**  This layer contains a DBMS which takes care about the data, their storage and their access. Only the Business tier can access to this tier.

here you can introduce the high level components of your architecture (in our basic example in the slides about design you find these in slide 7) and describe the main interaction between them (no details here. You can say why some components talk to each other, why, if the communication is synchronous or asynchronous, any other info you think is useful at this point).

1. **component view**

client, server→ ride manager,queue manager, distribution manager.

here you have a refinement of what you have in Section 4.B and identify sub-components. For instance, the diagram in slide 6 could be a diagram showing a  component view

1. **deployment view**: Definiscono le principali unità di distribuzione e le linee guida all’installazione.-->Si definiscono le tecnologie ed i linguaggi che verranno usati. Occorre verificare che ogni unità di runtime abbia una corrispondente unità di distribuzione.

    client,account factory,account manager, request,reserve,request manager

    client taxi,manage availability,manage request

    server

   this is what you have in slide 8, that is, the identification of the artifact that need to be deployed to have the system working

1. **runtime view**: Definiscono le unità di runtime (i componenti disponibili in esecuzione), mostrando come collaborano tra loro.-->Si verifica che per ogni funzionalità esista un’entità di runtime che le contenga.

this is what you have in slide 9 plus sequence diagrams describing the way components behave in order to accomplish a certain activity

1. **component interfaces**
2. here you define the interfaces of your components, that is, which operations they offer to the external world, their meaning, any input and output parameter (name, possible set of values/type)
3. **selcted architecturaal styles and patterns**

MVC, client server(3 tier)

are meant to include any explanation of the choices you have made and of their rationale.

1. **other design decisions**

design pattern?(maybe section 5)

pub/sub,singleton,

are meant to include any explanation of the choices you have made and of their rationale.

Section 5 contains the definition of any algorithm that you think it is important to describe for your system  
  
Section 6 and 7 contain a short explanation.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/