# 1. Introduction and Goals

Robot Inc. needs a website capable of taking rental orders for multipurpose robots that are in stock from consumers.

The main goals of this website are to manage stock, allow for reservation and to forward customers on to the payments portal for final booking.

Current robot stock is 10.

## 1.1 Requirements

| **Requirement** | **Description** |
| --- | --- |
| Authentication | Handle authentication from an in place SAML authentication system |
| Track stock | Check for availability of specific robots |
| Reserve stock | Reserve robot stock for customers. Handle abandoned reservations |
| Handle payments | Pass off to payment system for final payment before confirmation |

## 1.2 Quality goals

* The system must be able to handle 5 concurrent users
* The system must reliably handle payment transactions
* User information must be encrypted

## 1.3 Stakeholders

| **Role** | **Name** | **Expectations** |
| --- | --- | --- |
| CEO | Alan Lightfoot | Sign off functional changes |
| Architect | Aaron Aargon | Sign off technical design |
| Engineering Manager | Colin Candidate | Sign off technical design |

## 1.4 Constraints

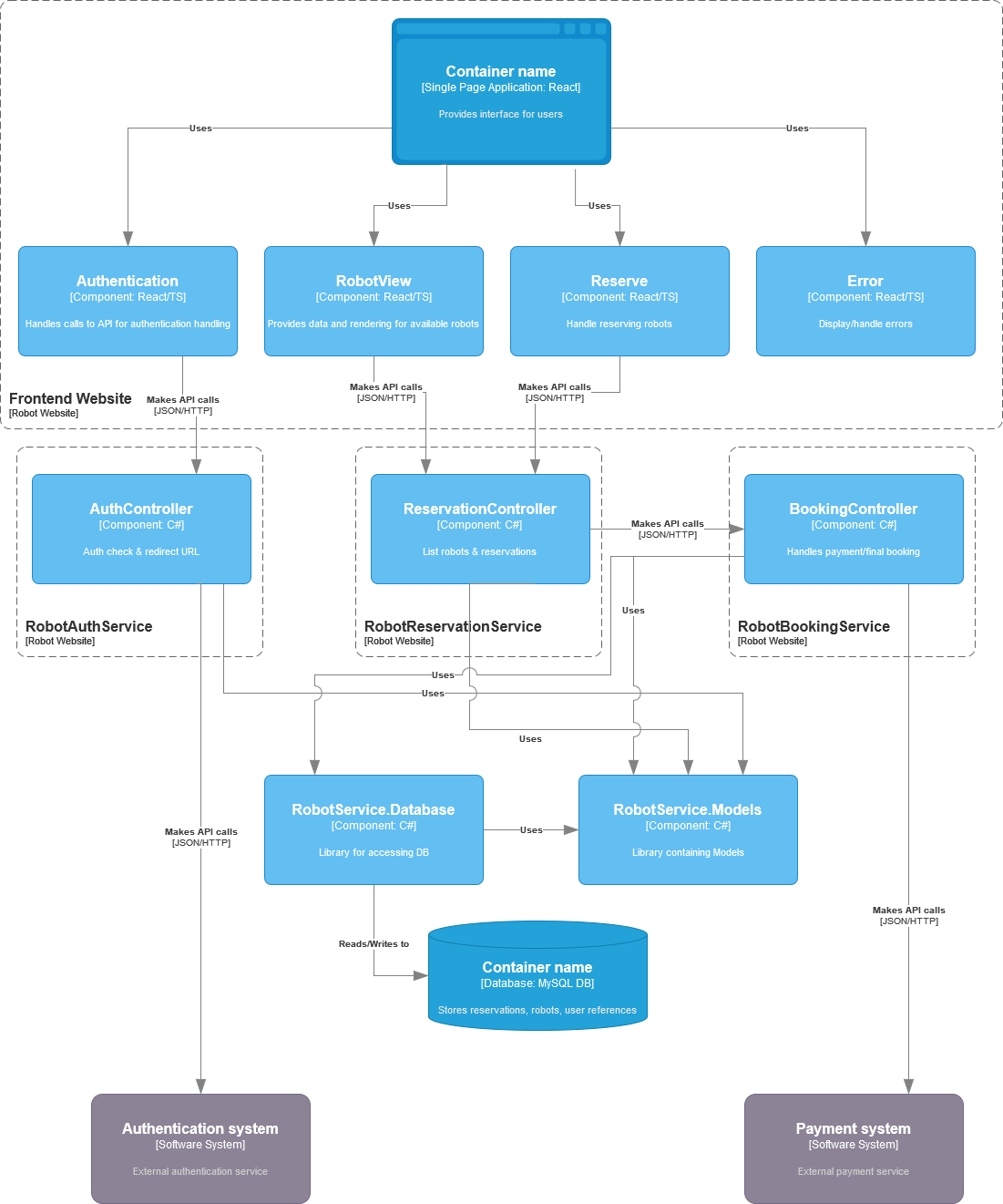
* Project must be delivered in 8 weeks ready for RobotCon
* Database must be run on PostgreSQL at sysops request
* Code must adhere to company standards

# 2. Solution Strategy

| **Scenario** | **Approach** | **Justification** | **Links** |
| --- | --- | --- | --- |
| Backend framework | C# .net 5.0. | Latest stable version, performs well and the team are experienced | https://dotnet.microsoft.com/ |
| Database | MySQL | Lead backend developer is very experienced in MySQL | https://www.mysql.com/ |
| Frontend framework | React | One of the developers said it's great | https://reactjs.org/ |
| ORM | Entity Framework | We’ll use EF to provide simplified access to the database | https://docs.microsoft.com/en-us/ef/ |
| Architecture pattern | Microservices | Other systems use this pattern | https://microservices.io/ |
| Service communication | TCP requests between services | Simple REST requests internally as low volume traffic |  |

# 

# 3. Runtime view



# 4. Persisted data structures

## MySql

### User

| Name | Type |
| --- | --- |
| Username | varchar(20) |
| Forename | varchar(20) |
| Surname | varchar(20) |
| Email | varchar(255) |

### Robot

| Name | Type |
| --- | --- |
| Name | varchar(20) |
| Type | int |
| Description | varchar(255) |

### Reservation

| Name | Type |
| --- | --- |
| Username | varchar(20) |
| RobotName | varchar(20) |
| StartTime | Timestamp |
| EndTime | Timestamp |
| Paid | BOOLEAN |

# 5. Components and libraries

## Frontend

### Components

| Name | Role | Scope |
| --- | --- | --- |
| Authentication | Handle authentication | * Redirect user to login * Provide logged in user information |
| RobotView | Provide details on available robots | * Provide information on available robots for a date range |
| Reserve | Handle reservation requests | * Check if robot is available on dates * Request reservation * Handle payment flow |
| Error | Handle errors | * Display errors in all workflows generically |

## Backend

### Services

| Name | Role | Scope |
| --- | --- | --- |
| Authentication | Provide handling of SAML auth | * Validate SAML data * Validate JWT tokens for authed customers |
| Reservation | Methods related to stock checking and reserving robots | * Check if robot is reserved on a date * Handle reservation of robot for customer * Communication with Authentication service to provide auth * Communication with Booking system |
| Booking | Handle payment and final booking | * Handle interaction with external payment system * Communicate with Reservation system to confirm or cancel reservation |

### Libraries

| Name | Role | Scope |
| --- | --- | --- |
| DB | Database access | * Common methods for communicating with DB |
| Models | Database models | * All models |

## 6. Overall concepts and approach

* Endpoints will be “deny” by default.
* Every user scenario will be covered by an end to end automated test
* Each endpoint will be tested for performance

# 7 . Risks and tech debt

No identified risks. Project is new, so no tech debt to address at this point.