Service Fabric Sample Application [Game]

1 Overview

This product is a ready-to-use sample application in the form of an online multiplayer game that an end-user can download and deploy on a local or remote cluster. The purpose is to give potential Service Fabric developers a sample application with lots of playability and learnability, motivating them to choose Service Fabric to build their multiplayer game on, and giving them the necessary understanding to get started. This sample application will differentiate from currently available ones because of a focus on:

(a) Implementation that plays to SF's strength in handling multiplayer games.

(b) Being a complete functional product serving a purpose.

(c) No expectation that the user has previous knowledge in the frameworks / libraries used.

2 Customer Scenarios

Customers are split to match the two experiences provided by the game: the first being the actual mechanism of gameplay, and the second is the use of the game as a learning platform for getting started on SF. Any given customer may experience both scenarios, as a casual player may want to see how the game works, and someone who wants to build on the application will likely deploy and play the game.

2.1 Casual Player Flow

(1) User goes to game webpage on their browser.

(2) User enters login credentials, which brings them to a room lobby that displays the currently active rooms. They can join a room or make their own.

  a. If the maximum number of rooms have been created, there is no "make room" option.

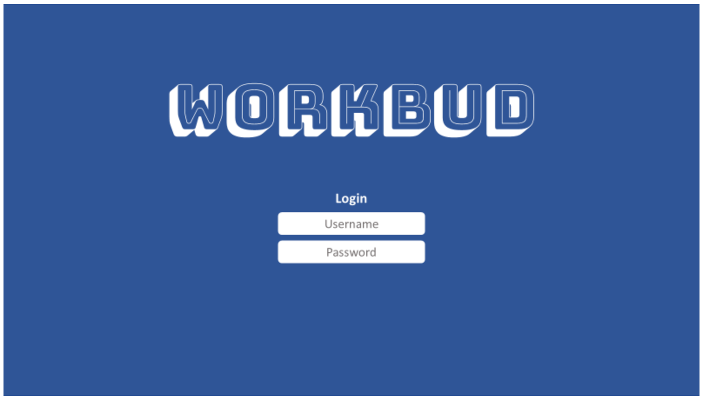
  b. If a room is full, it does not allow the user to join that room and tells them so.

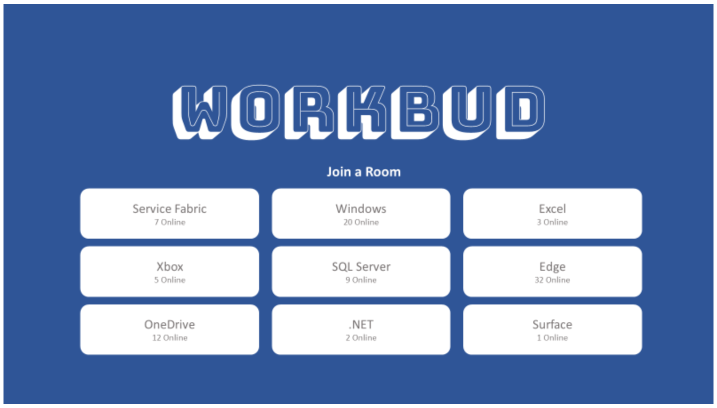
(3) Upon entering a room, they see every other player that is also logged into the room. They can move around, and those changes are visible to every other player.

(4) The user can choose to change features about themselves, such as character color. Those changes are persisted in their account and are visible to other players.

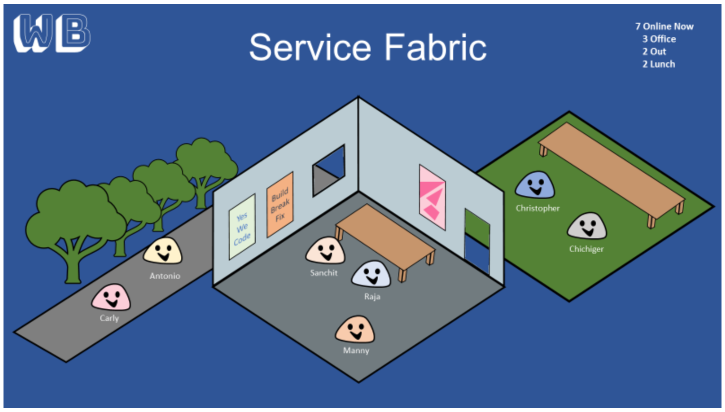
(5) Upon logging out or closing the browser window, they are removed from the room and removed from the other players' view.

(6) Upon relogging in they are welcome to join the same room or another room.





Service Fabric 7 Online	Windows 20 Online	Excel 3 Online
Xbox 5 Online	SQL Server 9 Online	Edge 32 Online
OneDrive 12 Online	.NET 2 Online	Surface 1 Online



2.2 Developer Flow

(1) User visits a website and downloads the game files.

(2) User deploys the game to their local host and can play game by accessing a web endpoint.

a. There is a document detailing how to deploy the game locally and set up a cluster.

(3) User reads through the game documentation, which helps them understand how everything works.

(4) From here the user can choose to

a. Deploy the application to the cloud, hosting a version of the game on a cluster. To assist with this there is a document detailing how to go about doing this.

b. Make changes to the game – adding or removing features. To assist with this there is a worksheet that leads them through the addition of a new feature to the game.

3 Requirements

3.1 Features

P0: MVP || P1: Complete Product || P2: Reach Product

	Casual Player Flow	Developer Flow
P0	Login page accepts user ids and persists the state associated with that user.	Every c# file in application is documented in XML-style comments.
	Lobby page shows player all active rooms and allows them to join or make a new room. They can choose a different room each time they log in.	Application comes with a document that details game architecture and how every component of the game fits together.
	Game allows player to see all the users in room and all their visible features.	
	Players can move around room and change their features – these changes are visible to all other players.	
P1	There is a sign-in page for new users and login uses persisted passwords.	Application comes with a document that details to user how to deploy the game to the cloud.
	When a user makes a room, they can choose among multiple room “settings” that change look of room.	Application comes with a worksheet that leads the user through adding another feature to the game.

	The user can engage in a game while in the room (e.g. tic-tac-toe).	
P2	Login page uses industry-standard encryption with account verification and password recovery.	Game components are packaged as libraries so that they can be used easily by game creators (e.g. service fabric login library, game state load balancer)
	Addition of implementation specific MMO features (e.g. web sockets, delay + animation smoothing).	

3.2 Metrics

Game runs nominally for up to 30 users per room and up to 30 rooms (900 users). Reach: 100 users each, in 100 rooms (10,000 users)
Time between a state change on one player’s client and another player seeing those changes in their client is less than a second.
In case of failover, in-game response is less than two seconds.

3.3 KPIs

UI is intuitive enough that it takes less than 30 seconds from going to website to be in-game.
Sample application is downloaded more times in the first month than the current average for this metric for service fabric sample applications.