# Software Requirements Specification

**for**

# Crew Connect

**Version 1.7 approved**

**Prepared by Jayna Celano**

**WASUP Airlines**

**2/9/2023**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| Jayna Celano | 2/15/2023 | Add in “Purpose” | 1.1 |
| Jayna Celano | 2/15/2023 | Add in “Document Conventions” | 1.2 |
| Jayna Celano | 2/15/2023 | Add in “Intended Audience and Reading Suggestions | 1.3 |
| Jayna Celano | 2/15/2023 | Add in “Product Scope” | 1.4 |
| Jayna Celano | 2/15/2023 | Add in “References” | 1.5 |
| Jayna Celano | 2/23/2023 | Changed title page to “WASUP Airlines” | 1.6 |
| Jayna Celano | 3/09/2023 | Add in “2.1-2.7” | 1.7 |

# 1. Introduction

## 1.1 Purpose

Crew Connect is the new application created for WASUP Airline. The updated version ___ of the application has been revised to create ease and convenience for all employees, i.e., users of the application. We aim to build an application that streamlines communication between corporate/flight operations and the crew. Through Crew Connect, we will solve issues regarding efficiency and effectiveness of the crew via increasing and simplifying communication during air routes.

## 1.2 Document Conventions

This document uses the following conventions:
Majority of generic font will be in Times New Roman, size 12, and in black font. Any notifications that maintain high priority will be highlighted yellow. Features that will need to be accessed without Wi-Fi connected will have a "*" next to it. Additionally, any vacation-related information will be placed in *italics*. Crew Connect may also be abbreviated as "CC" throughout the SRS.

## 1.3 Intended Audience and Reading Suggestions

| **User or Reader** | **Suggested Areas to Read** |
|---|---|
| Stakeholders | 1,2, 4 |
| IT, Developers, Software Architects | 1,2,3,4,5,6 |
| Project Manager | 1,2,4,5,6 |
| End Users | 1,2 |

## 1.4 Product Scope

Crew Connect was created to efficiently utilize employees and to simplify communication within WASUP Airlines. Through this software, the crew will be able to report vacations, receive notifications from corporate, request hotel changes, report general emergencies (i.e., sickness, hotels issues, etc.). Crew Connect will also allow corporate/flight operations to connect with the crew. Human resources will be able to assign vacation time, flight operations can submit scheduling, manager of travel can assign hotel and car services, and the information technology team can send in weather updates, through a multitude of easy-to-use features. These features will be catered to employee needs and urgencies, and many of which will be accessible regardless of Wi-Fi connection. In doing so, we will allow for WASUP Airlines streamline their communication and services, enabling them to save time and money and, therefore, increase overall profits.

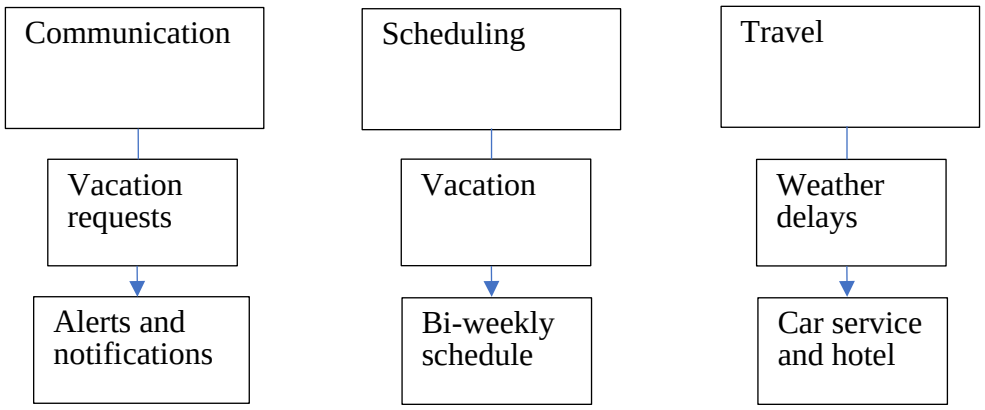
## 1.5 References

Visit www.wasupairlines.com for more information about current communication and management strategies. To learn more about this specific application (current issues, leads and staff, desired deliverables, etc.), visit the *Manual to Automated Crew Communications* guide sent via email by COO and sponsor, Scott Taylor on 1/23/23.

# 2. Overall Description

## 2.1 Product Perspective

CC is created as a new product, aiming to streamline communication between both ends of employees. This will be a new app for WASUP Airline that will be exclusive to the WASUP Airline.




## 2.2 Product Functions

CC will serve many functions for WASUP Airlines. However, the major functions involved in the design will surround these primary issues:

- Communication: ease communication between corporate and crew, including easy-to-use chats, notification and request systems, and alerts.
- Scheduling: show a clear depiction of bi-weekly schedules, vacation times, scheduling changes.
- Travel: show hotel and car accommodations, option for personal accommodations, notify of any weather delays or emergencies.

## 2.3 User Classes and Characteristics

There will be only one class of users using the CC application- the crew. While corporate will have access to the backend of the software, the crew will be actively using the application on their work-provided Android phones. This application will not be available to the public.

## 2.4 Operating Environment

The software will only operate using company supplied, Android hardware. This is because Android hardware has fewer bugs and limitations on corporate-based applications. Being that the software will only be used by the crew, there is no issue with coexisting components or applications.

## 2.5 Design and Implementation Constraints

There are certain constraints that will be necessary to implement within the CC software. For example, being that communication will exist between Human Resources and the crew, there will need to be limitations on the tables that can be accessed by table developers, as Human Resources has access to proprietary information (salaries, addresses, etc.). Additionally, there is the question of which languages need to be accounted for in the design. Will it be created using just English, or will it have options for Spanish, Chinese, etc.)? Also, an issue arises regarding who will be in charge of the maintenance, announcements, and training for usage of the software. Will WASUP Airlines oversee these tasks, or do we need to account for such activities?

## 2.6 User Documentation

In addition to the CC software, each user will be provided with an easy-to-follow online user manual. This manual will cover a broad overview of the application, further detail on each feature, and short videos on key aspects of the application. Moreover, there will be a user hotline available 24/7 to aid in any technical difficulties.

## 2.7 Assumptions and Dependencies

While CC is designed to be consistent and reliable, the application is still relying on other factors for its accuracy.

- Receives weather from weather applications.
- Receives delays from Air Traffic Control.
- Receives important travel information from hotels and car services.
- Dependent on the other systems we are pulling data from.

All of which could affect the accuracy of the application.

# 3. External Interface Requirements

## 3.1 User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## 3.2 Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## 3.3 Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## 3.4 Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# 4. System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## 4.1 System Feature 1

*<Don't really say "System Feature 1." State the feature name in just a few words.>*

#### 4.1.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

#### 4.1.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

#### 4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:
REQ-2:

### 4.2 System Feature 2 (and so on)

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## 5.2 Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>*

### 5.3 Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

### 5.4 Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

### 5.5 Business Rules

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

## 6. Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

## Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

## Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>*

## Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*