**1. Introduction**

**1.1. Purpose**

The purpose of this document is to detail the technical characteristics of the Keller-Williams Family Check-In (KWFCI) system by providing an overview for implementation and design.

Primary focuses of the KWFCI system include:

* Organizing the interactions necessary to keep connected to significant Keller-Williams brokers.
* Facilitate more effective communication through automated and web based tracking services.
* Provide feedback about interaction status through automated awareness alerts and notifications.

This document will not address every circumstance of user implementation or installation. When possible, clarification of intent and assumptions will be provided.

As development progresses this document may be adapted and updated to accommodate any changes in implementation.

**1.2 Scope**

The KWFCI will allow Keller-Williams staff members to sign in to a single interface.

The interface will have the ability to record broker interactions the staff enter into the system, with specified parameters being tracked.

The interface will provide notifications on status of interactions and be able to be used to refer to past recorded broker interactions.

Additional scope may vary according to the needs and requirements of the Keller-Williams staff, upon request.

**1.3 Document Organization**

This document is organized into the following sections:

|  |  |
| --- | --- |
| Introduction | Information related to this document. Includes purpose, scope, organization, and audience descriptions. |
| Design Overview | Defines the approach taken to implement the KWFCI system. This includes design goals and philosophies as well as technical details. |
| Use Cases | Details regarding the KWFCI usage by invested actors. |
| Application Architecture | Information explaining the design architecture of the KWFCI system. |
| Application Implementation | Directory structure and relevant details about how the KWFCI system is organized. |
| Database Architecture | Specifics of the database arrangement and architecture as it relates to the implementation of the KWFCI. |
| Assumptions and Constraints | A listing of assumptions made during the development process by the implementation team, as well as any constraints to be considered. |
| Appendix A | Acronym and abbreviation definitions. |
| Appendix B | Products and tools used. |
| Appendix C | Data dictionary of the system. |
| Appendix D | Configuration files required. |

**1.4 Audience**

The intended audience for this document include KWFCI design and implementation stakeholders, interface design team members, and technical guidance advisors.

**1.5 Acronyms and Abbreviations**

Appendix A contains a list of acronyms and abbreviations utilized in this document.

**2. Design Overview**

**2.1 Approach**

The development of the Keller-Williams Family Check-In (KWFCI) system will occur via a multiple phase process:

* *Requirements Phase* – an initial gathering of the necessities of the system through communication with the Keller-Williams staff, creative impressions on possible development from the implementation team, and utilizing similar established architectures to format an initial impression of requirements. First development of design document.
* *System Design Phase* – the implementation team refines development requirements into a working theory prototype. Prioritization of features is established to work within the time frame available for development. Design document changed in accordance with implementation adjustments.
* *Construction Phase* – refinement and beginnings of implementation for the system. The implementation team uses established guidelines to ensure progress towards the completion of the project. Limited design document adjustments, only in accordance with shifts of priority from Keller-Williams staff or unexpected necessary adaptations.
* *Implementation Phase* – quality assessment and training interactions between implementation team and Keller-Williams staff. Ensure that that project is fulfilling the requirements previously discussed to a level of satisfaction that the Keller-Williams staff is comfortable utilizing. Finalization of any changes to the design document.

**2.2 Architectural Goals and Constraints**

The intent of the KWFCI system is to provide an accessible and efficient organization interface for the Keller-Williams staff to keep track of communication between them and brokers which require specific personal interactions.

The KWFCI system will provide services to enhance organization, including, but not limited to:

* Recording significant details of specific communication events between staff and brokers.
* Providing alerts to bring attention to significant events that are time sensitive.
* Enable multiple brokers to be notified at once about future meetings or significant information.
* Sort and organize detailed information about communication exchanges to clarify prior exchanges.

The KWFCI system will be designed keeping scalability and region specific adaptations as possible future adjustment requirements. Implementation is designed to facilitate better organization of communication, but the majority of details of such communications are provided by Keller-Williams staff and subject to any errors that may occur during initial data entry. The scope of access to the KWFCI system will be dependent upon what hardware services the Keller-Williams staff decides to utilize for hosting.

**2.3 Guiding Principles**

The guiding principles for the KWFCI system provide a foundation for functionality and design. These provide a standard of quality to be met in order for a feature to be considered for implementation. The guiding principles provided here are in addition to design best practices and ethical standards which will be maintained by the implementation team.

**2.3.1 Functionality First**

The final goal of the KWFCI system is to provide organization and ease of understanding for the Keller-Williams staff to ensure that communication remains strong and positive between them and the brokers. With this in mind, aesthetic considerations will be taken, but only when the functionality of the system remains intact or enhanced by aesthetic changes.

**2.3.2 Scalable**

The intent of initial design for the KWFCI system is to provide an effective tool for local Keller-Williams staff. The scope of staff which can utilize the system has the potential to increase, so the system will be designed with this in mind. Local focus initially with the possibility of a broader implementation at a later date.

**2.3.3 Responsive Design**

The KWFCI system will provide efficient and productive feedback to the users of the system. Design parameters which would limit the functionality and responsiveness of the system will be adjusted accordingly.

**2.3.4 Easy to Use**

The intended users of the KWFCI system are the Keller-Williams staff who will be interacting with brokers. The design of the system will reflect the intended user audience and provide clear and meaningful interfaces to allow for helpful organization of communication between Keller-Williams staff and brokers. Ease of use is determined by input from the intended users of the system.

**2.3.5 Connecting KW Family**

The KWFCI system is intended to enhance and clarify interactions between Keller-Williams staff and brokers. If any part of the system is inhibiting or restriction the ability to allow communication between Keller-Williams staff and brokers it will be reassessed and modified accordingly.

**2.4 Design Patterns**

The implementation team will follow a Model, View, Controller (MVC) design structure when creating the KWFCI system. The MVC design structure will allow for separation of concerns for data identification, user interface, and system interaction. In addition, the implementation team will utilize design elements that effectively enhance the MVC design process.

The design structure elements specifically being focused on for implementation are:

* *Model* – the logic for the system's data.
* *View* – display of the system's user interface.
* *Controller* – component to handle system interactions.
* *Unit* *Testing* – tests to ensure proper functionality of system components.
* *Interfaces* – allow for wider definition of objects without immediate implementation.
* *Templates –* guided structures to ensure consistency.
* *DRY* *approach –* 'do not repeat yourself' as a means of enhancing system efficiency.

**2.5 Design Principles**

The KWFCI system is designed to be utilized through a web browser that can support a web application utilizing modern client-server communications for validation, processing, and display. The KWFCI system does not utilize a separate custom software to be able to be utilized and is intended to be accessible only via web browser software.

Any Keller-Williams business rules will be programmed into the system, as communicated by Keller-Williams staff to the implementation team. Support for keeping business rule implementation up to date is at the discretion of whoever the Keller-Williams staff employs to continue to provide maintenance for the KWFCI system after initial design.

**3. Use Cases**

<diagrams for admin, staff, broker>

**4. Application Architecture**

Describe application architecture

<diagram of MVC design>

describe the parts of MVC

<site map>

describe the views

<class diagram>

describe class properties/methods/inheritance

**5. Application Implementation**

<diagram of directory structure>

describe how the css etc is connected

**6. Database Architecture**

describe what type of database is being used

6.1 describe what a data model is and then <graphic of data model>

6.2 describing the tables of the data model, categorizing them into which groups they're relevant to

6.3 Querying solution example

**7. Assumptions and Constraints**

The design of the Keller-Williams Family Check-In (KWFCI) system is designed with specific constraints being considered. During the development the implementation team will make assumptions during the process, of which several are included below.

|  |
| --- |
| Assumptions and Constraints |
| Use of the system is designed to be applied via web browsers which are considered up to date as of 3/1/2017. Separate custom software will not be necessary to utilize the KWFCI. |
| The KWFCI system will be hosted at the expense of the Keller-Williams company so that it may be accessed using HTTP connections. Keller-Williams staff will be responsible for determining which services they would like to purchase to enable hosting services. |
| Maintenance of the KWFCI system will not be the responsibility of the implementation team beyond the initial establishment of the system. Further maintenance required for the system will be the responsibility of the Keller-Williams staff to determine an expert to manage the KWFCI. |
| Business rules, and other encoded specifications, will be established by the implementation team upon initial creation of the KWFCI. After initial implementation, it will be the responsibility of the Keller-Williams staff expert to make any changes to the KWFCI. |
| Initial scope of the KWFCI is intended for the Eugene area Keller-Williams company and any further expansions of the program to include application for other regions is considered beyond the scope of the initial development of the KWFCI. |
| Regular access to Keller-Williams staff, for the purpose of communicating clarifications, will be available for the implementation team. Response from the Keller-Williams staff will take no longer than seven days from the implementation team's initial inquiry. |

**Appendix A: Acronyms**

|  |  |
| --- | --- |
| AJAX | Asynchronous JavaScript and XML |
| CSS | Cascading Style Sheets |
| ERD | Entity Relationship Diagram |
| HTML | Hypertext Markup Language |
| KWFCI | Keller-Williams Family Check-In |

**Appendix B: Products and Tools**

<table of versions of software being utilized>

**Appendix C: Data Dictionary**

<table with information about fields from data dictionary>

**Appendix D: Configuration Files**

list some of the known basic required files (\*IF we even need this to get to page count)