Software Requirements Specification

**ChequeMeOut**

notsirkApps

**Prepared By**

|  |  |
| --- | --- |
| DOCUMENT OWNER(S) | Organization role |
| Kriston Sanders | Project Manager |

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change description |
| 1.0 | 5/28/20 | Kriston Sanders | Document created |
| 1.2 | 6/5/20 | Kriston Sanders | General revisions; removed superfluous customer class |
| 1.3 | 6/12/20 | Kriston Sanders | Additional requirements; formatting changes |
| 1.4 | 7/27/20 | Kriston Sanders | Additional requirements added |
| 1.5 | 7/30/20 | Kriston Sanders | Additional requirements added |

**Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Approving party | Signature |
| 1.2 | 6/5/20 | Kriston Sanders | Kriston Sanders |
| 1.3 | 6/12/20 | Kriston Sanders | Kriston Sanders |
| 1.4 | 7/27/20 | Kriston Sanders | Kriston Sanders |
| 1.5 | 7/30/20 | Kriston Sanders | Kriston Sanders |

| 1.0 Introduction |
| --- |
| * 1. **Purpose** * Reduce wait times, contact, and eliminate the need to split checks * Launch product that is fun to use and intuitive * Create product to highlight skills and impress potential employers * Be transparent about data collection/use * Accommodate transition between methods * Accommodate non-users * Low adoption cost   1. **Document Definitions** * Diner- customers of a restaurant or bar * User- a diner who utilizes the mobile application * Non-user- a diner who does not use the mobile application * Open diner pool- a subgroup in the Diner app to aggregate orders for non-users * Server- employee who interacts with the diners and tracks orders * Kitchen- employees who cook food * Owner- the owner of the bar or restaurant * Diner app- the mobile application used by diners to order and pay for food service * Terminal app- the mobile/desktop application used by servers, kitchens, and owners to track, settle, and monitor sales for orders * Shall- indicates a must-have requirement * Should- indicates a nice-to-have requirement.   1. **Intended Audience** * Development team * Recruiters   1. **Scope** * Customer mobile app for ordering and payment * Business applications for accepting and tracking orders and payments * Integrate with existing payment system (Square) * Secure data transmission * Store business and transaction data with Square or independent server * User authentication processed through Back4App   System will improve the dining out experience by reducing wait times, contact, and eliminate the need to split checks by offloading tasks to a suite of mobile applications. This system will bring in revenue through product licenses while bolstering my resume with a demonstration of my skills while also advancing them.   * 1. **References**   The ChequeMeOut Use Cases document clarifies these requirements. |

| 2.0 general description |
| --- |
| * 1. **Product Perspective**   Oftentimes diners are anxious to finish their meals and pay when they have somewhere else to be, such as a movie or Broadway musical, or simply when they are tired or are on a bad date. They also choose to dine elsewhere when there is a long wait for a table; however, if they do remain, they may have trouble distancing from others.  Restaurateurs have an interest in turning tables around to accommodate more paying guests, but the added wait times for ordering and settling the bill constrain this considerably.  When dining in groups, there can be quite a circus around splitting the bill and communicating that to the server. This requires calculators, cumbersome check-splitting apps , and attempts to split shared items, all of which can lead to bad calculations and the headache of writing this out or hoping the server remembers the numbers when cashing diners out. Often it is simpler to just split the check in half; however, some diners order more items or more expensive items that the other party(ies) must subsidize.   * 1. **Product Features** * Diners shall be able to order and pay from their phone * Check splitting shall be mostly automatic * Groups shall link to a table across their personal devices by scanning a QR code displayed at the table   1. **Operating Environment**   In restaurants and bars, running alongside traditional POS systems while industry transitions to new modes of operation.   * 1. **Constraints** * Job search may reduce work hours available * Training may be necessary, slowing down development * May be called back to work before end of project * May be pulled away for unemployment activities * No Macintosh for iOS simulating * Limited number of devices for testing   1. **Assumptions and Dependencies** * Most people carry a smartphone * Most people are willing to learn a new system if it saves time and hassle * Most restaurants have WiFi and adequate cellular coverage * Access to the Square API * Encryption package |

| 3.0 Functional requirements |
| --- |
| * 1. **Diner app Requirements** * App shall manage user authentication through Back4App * App shall check for registration upon launch and collect registration information, including email address, name, address, and credit card details if not on file * App shall pass user and payment details to Square securely * App shall link users together to a table * App shall display a menu that users can order from * App shall allow users to place orders for non-users and place the items in the open diner pool * App shall allow users to modify their cart before submission * App should allow users to request cancellation of items that have been submitted * App shall notify users when it is time to settle a table and allow them to select a tip about * App shall allow users to split shared items with other users or the open diner pool * App should allow users to change number of shares per user on shared items * App shall submit orders to the Terminal app * App shall allow users to settle their portion before the table as a whole * App shall allow users to increase their tips after settlement * App should allow users to place items under consideration (flag them) and to select from this list * App shall allow users to order a non-menu item (special) * App should allow users to call servers to their table * App should allow users to offer to pay for other’s orders * App should prompt users for additional details on certain items, such as liquor brand for drinks * App should support spice levels for Asian and other restaurants * App should allow users to change font sizes * App shall accommodate colorblindness * App should support free refills for drinks * App shall prompt user for feedback when leaving a low tip * App should allow users to upload a photo for their profile   1. **Terminal app Requirements** * App shall have server mode, kitchen mode, and owner mode * App shall allow servers to take ownership of established tables * App shall allow servers to pass off a table at shift change * App shall notify servers assigned to a table has orders are submitted * App shall allow servers to submit orders to the kitchen at will so that orders in a large group can be submitted simultaneously or piecemeal * App shall receive orders for the kitchen * App shall allow servers to mark items delivered * App shall periodically notify servers of undelivered orders * App should notify servers of cancellation requests and allow them to remove items * App shall allow servers to settle tables * App shall indicate early settled portions of the bill * App shall allow owners to monitor transactions and transaction history * App shall allow owners to settle a day/week to gather income data and export tip data to their payroll system * App shall allow server to edit non-menu items (specials) * App should allow servers to prompt diners for additional details, such as liquor brand for drinks * App should allow servers to make doodles on settlement requests * App should allow owners to customize menu formats, such as with HTML * App should allow staff to assign tables to servers at shift start and be changeable at any time * App shall accommodate colorblindness * App should support free refills for drinks * App should display user photo to help identify where to deliver orders   1. **System Requirements** * System shall support:   + iOS 12.x and above   + Android 9.x and above   + UWP on Windows 10 * System should support:   + - Android 8.x     - x86 on Windows 10   + System shall store user/table/transaction data Square * System shall manage secure communication between Diner and Terminal apps through server * System should support additional payment options such as:   + Apple Pay   + Google Wallet   + PayPal   + Venmo |

| 4.0 Non-functional requirements |
| --- |
| * 1. **Performance Requirements** * Software shall be intuitive and easy to use * Apps shall have very low crash rates * Apps shall feel speedy and smooth * Apps shall not contain superfluous pages or steps   1. **Security Requirements** * System shall transmit data securely * System shall not store user/payment information except where necessary * Apps shall use authentication * App shall allow owners to protect transaction history with a password   1. **Software Quality Attributes** * Software shall be subject to continuous integration and test-driven development * Code shall be developed adhering to object-oriented design, with low coupling and high cohesion * Code shall be highly extensible and easy to read   1. **Other Requirements** * Square API use shall adhere to the user agreement unequivocally * Projects will be developed using Visual Studio with Xamarin in C# * Project management will be done in Azure DevOps * Azure pipelines shall be used to test code continuously * Code will be stored in a Git repository * Apps shall be readable and usable by those with colorblindness * Apps shall not require hearing to operate * Apps should be usable with alternate input types * Apps should be usable by the blind |