**CHEXMAN**

**(CHARGEABLE EXPENSES MANAGEMENT SYSTEM)**

**ESPECIFICACIÓN DE REQUERIMIENTO DE LA Aplicación**

**AUTOR - GRUPO**

**THE COOL CLUB COLLECTIVE**

**Contenido:**

1.Introducción

2.Objetivos

- documento de requerimiento

3.Requisitos de la aplicación

- temas legales

4.Requerimientos

- funcionales

- no funcionales

5.Analisis y diseño

- ciclo de vida de modelo de desarrollo

- codificación de la App

- diseño de la base de datos

- página web presentación aplicación

6.Especificación funcional

- casos de uso

- diagramas

- mapa de relaciones

7.Especificación técnica

8.Seguridad

**1.Introducción:**

La tecnología avanza constantemente y nuestra empresa el Grupo The Cool Club Collective se ha visto en la necesidad de buscar herramientas que las ayuden a mejorar sus flujos de trabajo y aumentar la satisfacción de los clientes.

Una empresa que esté direccionada hacia el éxito debe ser eficiente en todos los sentidos. Al final, las transformaciones son necesarias para adaptarnos a las nuevas rutinas. Lo cierto es que ya no estamos en la era de los productos y servicios, sino en la era de las experiencias.

 Chexman es una aplicación de ticketing de uso interno de cualquier tipo empresa. Es una App centrada en el soporte de software o hardware para facilitar la relación entre empleado-RRHH y minimizar y controlar las posibles incidencias. Esté software te permitirá no olvidar ninguna de las incidencias y consultas creadas por parte del trabajador o RRHH en una empresa y mantener el trabajo ordenado.

**2.Objetivos:**

**2.1.** El propósito de esta Aplicación es mantener, optimizar y organizar de una forma más sencilla y fácil la relación entre los empleados y RRHH de una empresa.

**2.2.** Suministrar un historial de forma clara para explicar el funcionamiento de la App.

**2.3.** Gestionar de una manera fácil las incidencias y el tiempo invertido del soporte técnico a solucionar las.

**3.Requisitos de la Aplicación - Temas legales:**

**3.1. Permisos, licencia y condiciones de uso**  
Hay que ser claros y explícitos a la hora de solicitar permisos al usuario para acceder a contactos de su dispositivo, realizar pagos o ceder datos. Además,**es obligatorio desarrollar licencias y condiciones de uso.** En todos los casos no basta con informar al usuario, sino que este tiene que aceptar, puesto que en caso de reclamación tendremos una mejor defensa.  
  
**3.2. Derechos propios y de terceros**  
Es obligatorio disponer de licencias de los recursos que se utilizarán. Para lo cual, hay que leer detenidamente las condiciones, puesto que hay casos en los cuales los recursos excluyen el uso comercial, no pudiéndose ejecutar en aplicaciones. Además, **conviene proteger el contenido para evitar plagios y copias.**  
  
**3.3. Menores**  
En caso de aplicaciones dirigidas a menores de 14 años, **se tienen que consultar las leyes correspondientes y las obligaciones impuestas**. El motivo es que existe una regulación especial en materia de consumidores y usuarios, protección de datos, derechos de imagen, etc.  
  
**3.4. Funcionalidades lícitas**  
Igual que en el marketing tradicional,**el que es ilícito al offline, a la app también lo es**. Por ejemplo, estimular un ámbito de vida poco saludable, como el consumo excesivo de alcohol u otras sustancias.  
  
**3.5. Privacidad y geolocalización**  
La recogida de información del usuario tiene que ser la indispensable para el funcionamiento de la lo app, y este tiene que tener la posibilidad de configurar la privacidad. Además, **si nuestra aplicación dispone de geolocalización, se tiene que contar con la aceptación del usuario** para poder acceder a ella.  
  
**3.6. Información y cookies.**  
Es **fundamental informar el usuario de los aspectos regulados en la ley** y mostrar los datos sobre los creadores y sobre quienes hay detrás de la aplicación. También es necesario que el usuario acepte las cookies mediante un aviso informativo con la información básica y precisa sobre las mismas, así como los aspectos exigidos por la ley.  
  
**3.7. Markets**  
**Tienen condiciones muy estrictas porque se puedan publicar las aplicaciones. Por lo tanto, hay que cumplir siempre el que piden.** De hecho, incluso cumpliendo las condiciones al colgar la app, estas pueden cambiar y hacer que la aplicación no esté disponible para usuarios nuevos. Un ejemplo que suelen alegar los markets para rechazar una app es que su interfaz de usuario es compleja o inferior al que catalogan como "muy buena".  
  
**3.8. Publicidad**  
Si monetizan una aplicación a través de publicidad, esta tiene que identificarse siempre como tal.  
Se **recomiendan a los emprendedores que no pasen por alto la normativa legal** puesto que "el incumplimiento de estos requisitos puede generar problemas con la Administración, la justicia y los usuarios. Además, de poder provocar retrasos en el lanzamiento, las actualizaciones o modificaciones de última hora de la aplicación que no estaban previstos y que pueden suponer un coste añadido al proyecto", aseguran.

**3.9. La ley de protección de datos**

**- La Ley Orgánica 15/1999 de 13 de diciembre de** **Protección de Datos de Carácter Personal, (LOPD),** es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Su objetivo principal es regular el **tratamiento de los datos y ficheros de carácter personal**, independientemente del soporte en el cual sean tratados, los **derechos de los ciudadanos** sobre ellos y las **obligaciones de aquellos que los crean o tratan**.

**Desarrollo normativo**:

* **El Real Decreto 994/1999 de Medidas de Seguridad de los ficheros** automatizados que contengan datos de carácter personal de 11 de junio de 1999 (RMS) : Es un reglamento que desarrolla la Ley Orgánica 5/1992, de 29 de octubre, de Regulación del Tratamiento Automatizado de los Datos de Carácter Personal (LORTAD), regula las medidas técnicas y organizativas que deben aplicarse a los sistemas de información en los cuales se traten datos de carácter personal de forma automatizada.(Derogado desde 19 de abril de 2010)
* **El Real Decreto 1720/2007, de 21 de diciembre de desarrollo de la Ley Orgánica de Protección de Datos. Se trata de un desarrollo de la Ley Orgánica 15/1999 de Protección de Datos de 13 de diciembre;** desarrolla tanto los principios de la ley, como las medidas de seguridad a aplicar en los sistemas de información. Se aplica tanto a ficheros en soporte automatizado, como en cualquier otro tipo de soportes.

**Obligatoriedad de la Ley**:

Esta ley obliga a todas las personas, empresas y organismos -tanto privados como públicos que dispongan de datos de carácter personal- a cumplir una serie de requisitos y aplicar determinadas medidas de seguridad en función del tipo de datos que posean.

A grandes rasgos las obligaciones legales fundamentales son:

* Dar de alta los ficheros en la Agencia Española de Protección de Datos.
* Elaborar y mantener actualizado el Documento de Seguridad.
* Obtener la legitimidad de los afectados.

**Régimen de control**:

El órgano de control del cumplimiento de la normativa de protección de datos dentro del territorio español, con carácter general es la **Agencia Española de Protección de Datos (AEPD)**, existiendo otras Agencias de Protección de Datos de carácter autonómico, en las Comunidades Autónomas de Madrid, Cataluña y en el País Vasco.

Las **sanciones** tienen una elevada cuantía, siendo España el país de la Unión Europea que tiene las sanciones más altas en materia de protección de datos.

**4.Requerimientos:**

**4.1. Requerimientos funcionales:** Poder acceder y gestionar fácil los datos y los gastos de los empleados

|  |  |  |
| --- | --- | --- |
| Índice | Titulo | Descripción |
| 1 | Login - Entrada | Entrada en el sistema mediante usuario y contraseña |
| 2 | Logout - Salida | Salir del sistema |
| 3 | Alta empleado | Crear los registros de todos los empleados de la empresa |
| 4 | Crear ticket | Crear los registros de todos los tickets de la empresa ¿? |
| 5 | Añadir ticket | Añadir o relacionar un ticket con un empleado |
| 6 | Buscar empleado | Mostrar el registro de un empleado que atienda la empresa mediante su identificación |
| 7 | Buscar ticket | Mostrar el registro de un ticket mediante su identificación |
| 8 | Borrar empleado | Borrar el registro de un empleado |
| 9 | Borrar ticket | Borrar el registro de un ticket |
| 10 | Actualizar empleado | Actualizar el registro de un empleado |
| 11 | Actualizar ticket | Actualizar el registro de un ticket |
| 12 | Actualizar empleados | Actualizar la lista con todos los empleados de la empresa |
| 13 | Actualizar tickets | Actualizar la lista de todos los tickets existentes en la empresa |
| 14 | Buscar empleado-ticket | Mostrar una lista con todos los empleados y los tickets que cada empleado ha creado |
| 15 | Buscar incidencias | Comprobar que el sistema funciona bien |

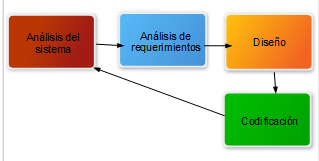
**4.2. Requerimientos no funcionales:**

|  |  |  |
| --- | --- | --- |
| **Índice** | **Título** | **Descripción** |
| 1 | Multiplataforma | La aplicación se podrá ejecutar como mínimo en un PC, en un sistema running IOS y en un sistema running Linux Debian o Linux Elementary OS. |
| 2 | Mantenimiento | Separar la implementación de la interfaz de usuario de la de proceso de datos e implementar las funciones de acceso y proceso de datos en la base de datos siempre que sea posible. |
| 3 | Seguridad | Los usuarios del sistema tendrán acceso solamente a las partes de la aplicación y a las entidades de la base de datos que sean necesarias para cumplir con los requerimientos. Estos usuarios no tendrán acceso a las funciones generales y de administración de la base de datos misma. |
| 4 | Integridad relacional | Utilizar la tecnología de MySQL para asegurar que operaciones que acceden a más de una base de datos se realicen de manera que mantengan la integridad relacional. |

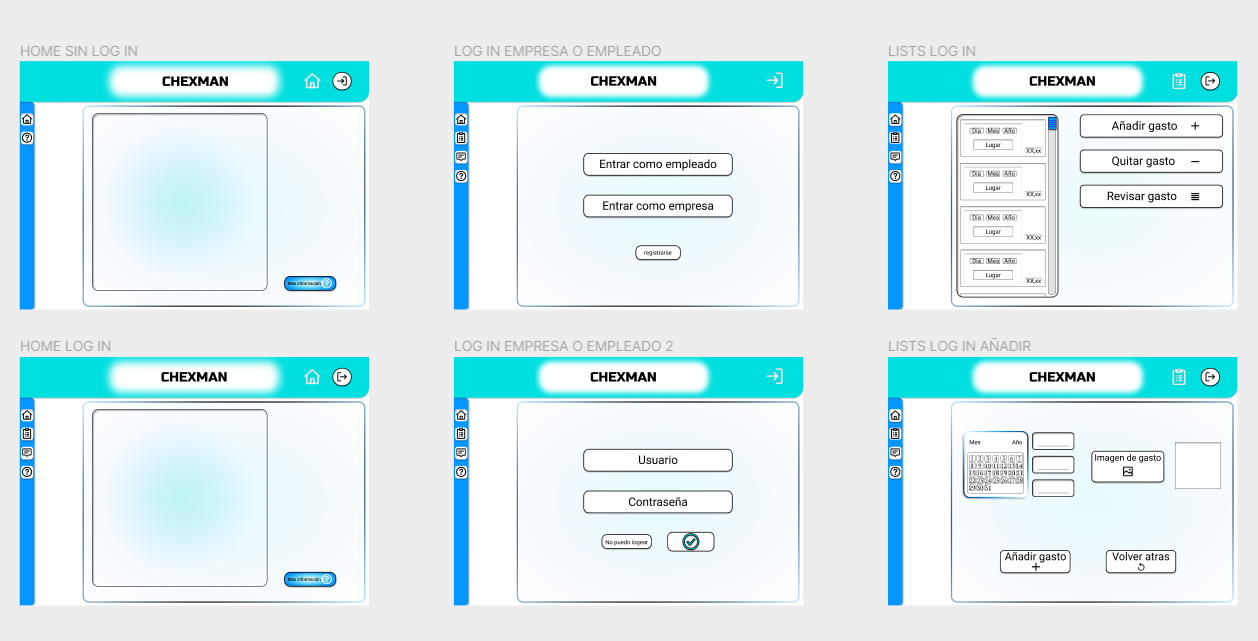
**5.Analisis y diseño:**

  En este punto, el enfoque está dirigido a reunir toda la información para el problema planteado anteriormente, y así determinar todos los requisitos funcionales y no funcionales, a partir de esto se desarrollarán las especificaciones de diseño.

**5.1**. **Ciclo de vida de modelo de desarrolló**



1. **Análisis del sistema:** En esta etapa se definirán las necesidades del cliente que presenta para la creación de la aplicación de tickets.



**b) Análisis de requerimientos:** En esta fase, se definirán las funcionalidades del sistema que deberían tener los mismos. Los cuales deben estar bien definidos para evitar un posible retroceso en los procesos siguientes.

**c) Diseño:** Esta fase está compuesta por varios factores, los cuales permitirán identificar las clases y los diagramas usados, y así definir la arquitectura y el motor de la base de datos usados, permitiendo diseñar las interfaces, basándose en las funcionalidades de la aplicación que estaban planteadas y analizadas al principio del proyecto.

**d) Codificación:** En esta fase, todo lo planeado en el diseño anterior se continuará con la codificación y la estructura de la base de datos, basado en los diagramas usados.

**Base de datos MySQL Con Tablas:**

* Login – Logout: - usuario; contraseña; código-empleado; código-soporte técnico; código-RRHH; código-Admin; fecha-hora-entrada; fecha-hora-salida
* Empleados: - código-empleado; DNI/NIE; nombre; apellidos; dirección; CP; Localidad; Provincia; email; teléfono; nomina
* Tickets: - código-ticket; fecha-crear; fecha-cobrar; cantidad; tipo-ticket; estado
* Tipo-tickets: - horas-extra; viajes; comida
* Estado-tickets: - en-espera; en-tramite; pagado; con-incidencia
* Incidencias: - código-ticket; código-soporte técnico; tipo-incidencia; fecha-incidencia; tiempo-resolver
* Tipo-incidencias: software; hardware; caída-servidor; error-empleado; error-rrhh; otros
* Soporte técnico: - código-técnico; nombre; apellidos; RRHH: - código-rrhh; nombre; apellidos; DNI; dirección; teléfono; sueldo
* Admin-jefe: - código-Admin
* Supervisar-1,2:
* Teléfonos-empleados: - fijo; móvil; de trabajo

**6.Especificacion funcional:**

- Los Actores son:

1. Admin – jefe de la empresa

2. Admin – RRHH

3. Soporte técnico

 4. Empleado

Todos los requerimientos son aplicables a los usuarios con privilegio de administrador.

A los usuarios de los actores RRHH y el soporte técnico con privilegios de administrador con restricciones.   
A los usuarios de actor Empleado solo login, logout y requerimientos específicos de empleado son aplicables el caso de uso del empleado son siempre una funcionalidad restringida de un caso general disponible para el administrador.

**6.1. Casos de uso:**

Entrada

Salida

Admin-jefe Admin-RRHH

Alta empleado, ticket

Buscar empleado, tricket

Modificar empleado,ticket

Actualizar empleado,ticket

Borrar empleado, ticket

Supervisar todos empleados, RRHH, ticket, soporte técnico

Empleado

Salir

Entrar

Buscar ficha empleado

Buscar ficha ticket-empleado

Entrada

Salida

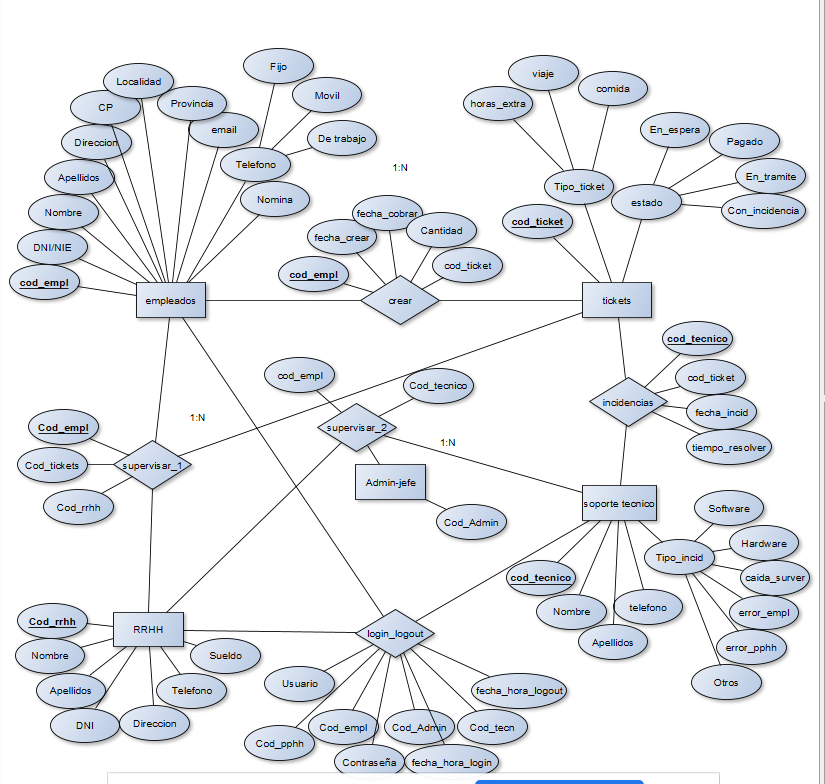
Soporte-técnico

Buscar y Arreglar incidencia

Modificar el sistema

Actualizar el sistema

**6.2. Diagramas**



**6.3. Mapa de relaciones:**

**7.Especificación técnica:**

El sistema se distribuirá como una aplicación web que puede ser ejecutada en cualquier plataforma.

Se ha decidido que se desarrolle con NetBeans usando servlets para acceder a las funciones de la base de datos y jsp paginas para proveer la interfaz para el usuario.

El sistema donde va a residir la aplicación deberá tener instalado MySQL donde se pueda instalar y ejecutar la base de datos, así como un servidor de aplicaciones (recomendamos Tomcat) donde instalar y ejecutar los servlets.

Hay un Servlets por requerimiento para ayudar con la trazabilidad de los requerimientos y el mantenimiento de la funcionalidad.

Todos los accesos a la base de datos se hacen por medio de procedimientos almacenados para facilitar la extensión y el mantenimiento. Los procedimientos almacenados que realicen labores de borrado solo borrarán la tabla principal, todas las operaciones de borrado adicionales se harán por medio de triggers. Este diseño cumple con el requerimiento no funcional Número 4.

Para la realización de pruebas y para la instalación se provee un conjunto de scripts que generan la base de datos, las tablas, los procedimientos almacenados y los triggers, así como un script que provee datos de pruebas para popular base de datos.

**8.Seguridad:**

La base de datos contiene datos personales que están protegidos por legislación vigente, por lo tanto, la seguridad debe esforzarse directamente en la base de datos. Restringir acceso solo por medio de la aplicación no cumpliría con la legislación.

El administrador de la base de datos será responsable de crear usuarios nivel administrador y nivel profesor para todos los usuarios del sistema.

Los usuarios nivel administrador tendrán todos los privilegios sobre el contenido de todas las tablas en este sistema, pero no tendrán privilegios de cambiar estructura o hacer “drop” en las tablas.

Los usuarios nivel RRHH tendrán privilegios de elegir y ejecutar solo procedimientos almacenados que contienen funcionalidad que cumple los requerimientos específicos de este tipo de usuario.

Cuando se crea un profesor a través de la aplicación es necesario insertar el nombre del usuario asignado para que el sistema sea capaz de acceder a las funciones de RRHH. Se puede crear un RRHH sin usuario, pero este no sería capaz de acceder a ninguna de las funciones del sistema hasta que esta información se haya completado en la tabla de RRHH.

Esta especificación asume que la tecnología MySQL provee las funciones necesarias para configurar la base de datos como está especificado. Este es un supuesto razonable, pero no se ha comprobado antes de la publicación de este documento.