

技术方案

视觉识别方案

视频采集设备

- 视频采集设备数量为 **两个**，方向为正视前方道路和后置勘察道路。
- 后置相机素质低于前置。

由于本系统需要在运动过程中进行视频采集，并对与视频数据进行分析判断，我们选用了较高帧数的摄像头，同时考虑到系统需要在夜晚缺少光源的情况下进行使用，该摄像头需要具备红外摄像功能。（视频传输质量为 **1920*1080@60fps**）

- 具体的型号参数如下：

产品参数

Modle.NO	星光级低照度摄像头		
感光型号 Sensor	IMX290		
模组尺寸 Modle Size	38mmx38mm±1mm（可根据客户要求定制）		
感光尺寸 Lens Size	1/2.8"		
像素面积 Pixel Size	2.9µm×2.9µm		
工作温度 Temperature（Operation）	'-20℃ to 70℃		
稳定温度 Temperature（Stable Image）	0℃ to 50℃		
生产工艺 Assembly technique	SMT（ROSH）		
成像距离 Object distance	1CM-∞（ 跟镜头有关 ）		
模块颜色 PCB printing ink	Black		
标配接口 Interface	USB3.0 A型母口端子		
线材长度 Wire length	1米（ 标准出货线长：1米 ）		
供电方式 Power	USB bus power		
工作电流 Power system requirements	160mA（VGA）		
工作电压 Operating Voltage	USB-DC5V		
操作系统 Operating system request	Windows XP/7/8/10，支持UVC协议的Linux，Android(支持UVC协议)，MAC-ios等		
免驱协议	USB Video Class（UVC）		
可调参数	亮度、对比度、色调、饱和度、清晰度 伽玛、白平衡、逆光对比、增益、曝光值		
自动曝光 AEC	支持(默认出货：自动曝光)		
自动白平衡 AEB	支持（ 默认出货：自动白平衡 ）		
最低照度	0.0001Lux		
压缩格式 Lmage compression	MJPG/YUY2(默认出货：MJPG)		
分辨率帧率 Active Array Size Video Rate	分辨率	MJPG帧率	YUY2帧率
	640x480	60fps	60fps
	1280x720	60fps	60fps
	1920x1080	60fps	60fps

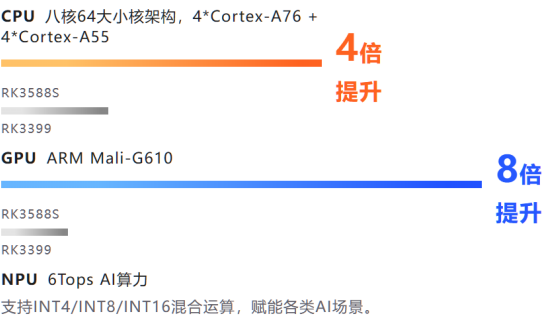
- 同时为了摄像头的稳定性与成像的清晰度，我们为摄像头都增加了云台设备，初步选定的云台设备型号为：**FeiyuTech G6 Plus**
- 在目前的基础上我们可以得到一个稳定且清晰的视频采集设备。

视频处理与分析设备

本系统使用的视频采集设备的高清与高帧数素质，我们就需要有更强的视频处理能力，我们的**视觉板**就需要有更高的性能要求，就简单的计算来看 **Raspberry Pi 4 Model B** 很有可能无法满足我们对于系统 **视觉板** 的要求，加之我们需要使用基于 **深度学习** 的AI大模型识别算法，我们选择了更加适合的 **Orange Pi 5 Pro**，它的 **ARM Mali-G610** 架构芯片对于AI能力大幅度提升。



性能巨大飞跃



- 视觉板参数介绍：

主控芯片	Rockchip RK3588S (8nm LP制程)
CPU	<ul style="list-style-type: none">8核64位处理器4核Cortex-A76和4核Cortex-A55的典型大小核架构，大核主频2.4GHz，小核主频1.8GHz
GPU	<ul style="list-style-type: none">集成ARM Mali-G610内置 3D GPU兼容 OpenGL ES1.1/2.0/3.2、OpenCL 2.2 和 Vulkan 1.2
NPU	内置高达6 Tops算力的AI加速器NPU，支持INT4/INT8/INT16混合运算
PMU	RK806-1
RAM	4GB/8GB/16GB/32GB (LPDDR4/4x)
存储	<ul style="list-style-type: none">QSPI Nor FLASH 16MBMicroSD (TF) Card SlotM.2 M-KEY Socket
USB	USB3.0 * 1; USB2.0 * 2; Type-C (USB3.1) * 1
视频输出	<ul style="list-style-type: none">HDMI2.1，最大支持8K @60HzDP1.4 (DisplayPort)，最大输出分辨率可达 8K@30Hz2 * MIPI D-PHY TX 4Lane，最大可配置为 4K @60Hz
摄像头	<ul style="list-style-type: none">MIPI CSI 4Lane2 * MIPI D-PHY RX 4Lane
音频	<p>CODEC:ES8388</p> <ul style="list-style-type: none">3.5mm 耳机孔音频输入/输出输入: 板载MICHDMI 2.1 eARC
以太网	10/100/1000Mbps以太网
扩展口	用于扩展UART、PWM、I2C、SPI、CAN和GPIO接口
M.2 M-KEY Socket 扩展槽	<p>支持PCIe NVMe SSD 固态硬盘</p> <p>支持定制PCIe Wi-Fi6+BT5.0模块</p>
按键	1* MaskROM键, 1*RECOVERY, 1 * 开关机键
供电	<ul style="list-style-type: none">支持Type-C供电, 5V @ 4A
LED	电源指示灯: 红色;状态指示灯: 绿色
调试	3Pin调试串口 (UART)
支持的操作系统	OrangePi OS (Droid)、OrangePi OS (Arch)、OrangePi OS (OH)、Ubuntu、Debian、Android12

后续的优化方案：

- 将视觉板更换为 **OpenHarmony** 支持的系统开发板 **Unionpi Tiger**，方便于接入整个 **Harmony** 生态以后将数据流通于整个生态之中，做到真正的 **车机互联**，**全方位信息整合**。

视频处理方案

目前针对于此视频输入的处理分为以下几个步骤：

- 原始视频图像初步处理
- 视频图像的识别与判断

原始视频图像处理

- 视频防抖算法：

```
import cv2
import numpy as np

# 打开视频文件
cap = cv2.VideoCapture('input_video.mp4')
```

```

# 获取视频的宽度和高度
width = int(cap.get(3))
height = int(cap.get(4))

# 创建一个空白的帧用于存储平滑后的图像
smoothed_frame = np.zeros((height, width, 3), dtype=np.uint8)

# 创建一个滑动窗口的大小（窗口越大，平滑效果越明显）
window_size = 15

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # 将当前帧与平滑后的帧相加，以实现平均平滑
    smoothed_frame = cv2.addWeighted(smoothed_frame, 0.8, frame, 0.2,
0)

    # 显示平滑后的帧
    cv2.imshow('Smoothed Video', smoothed_frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# 释放视频捕获对象和关闭窗口
cap.release()
cv2.destroyAllWindows()

```

- 简单滤波与处理反光：

```

import cv2
import numpy as np

# 打开视频文件
cap = cv2.VideoCapture('input_video.mp4')

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # 将图像转换为灰度图像
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # 使用均值滤波器对灰度图像进行平滑处理
    blurred_frame = cv2.blur(gray_frame, (15, 15))

    # 从平滑图像中减去原始图像，以减少反光效应
    result_frame = cv2.subtract(gray_frame, blurred_frame)

    # 显示处理后的帧

```

```

cv2.imshow('De-Reflection Video', result_frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# 释放视频捕获对象和关闭窗口
cap.release()
cv2.destroyAllWindows()

```

视频的识别与判断

这一部分的识别将会极其的复杂与繁琐，我们选择使用 **TensorFlow** 框架下来进行模型训练，并将前一步处理完成的图像交由 **AI模型** 来进行具体的决策和分析。

以下是一个简单的道路识别示例：

```

import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt

# 导入和准备数据集（这里使用CIFAR-10作为示例数据集）
(train_images, train_labels), (test_images, test_labels) =
datasets.cifar10.load_data()

# 将像素值缩放到0到1之间
train_images, test_images = train_images / 255.0, test_images /
255.0

# 定义卷积神经网络模型
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32,
32, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10)
])

# 编译模型
model.compile(optimizer='adam',

    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])

# 训练模型
history = model.fit(train_images, train_labels, epochs=10,

```

```
validation_data=(test_images, test_labels))
```

```
# 评估模型性能
```

```
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
```

```
print('\nTest accuracy:', test_acc)
```

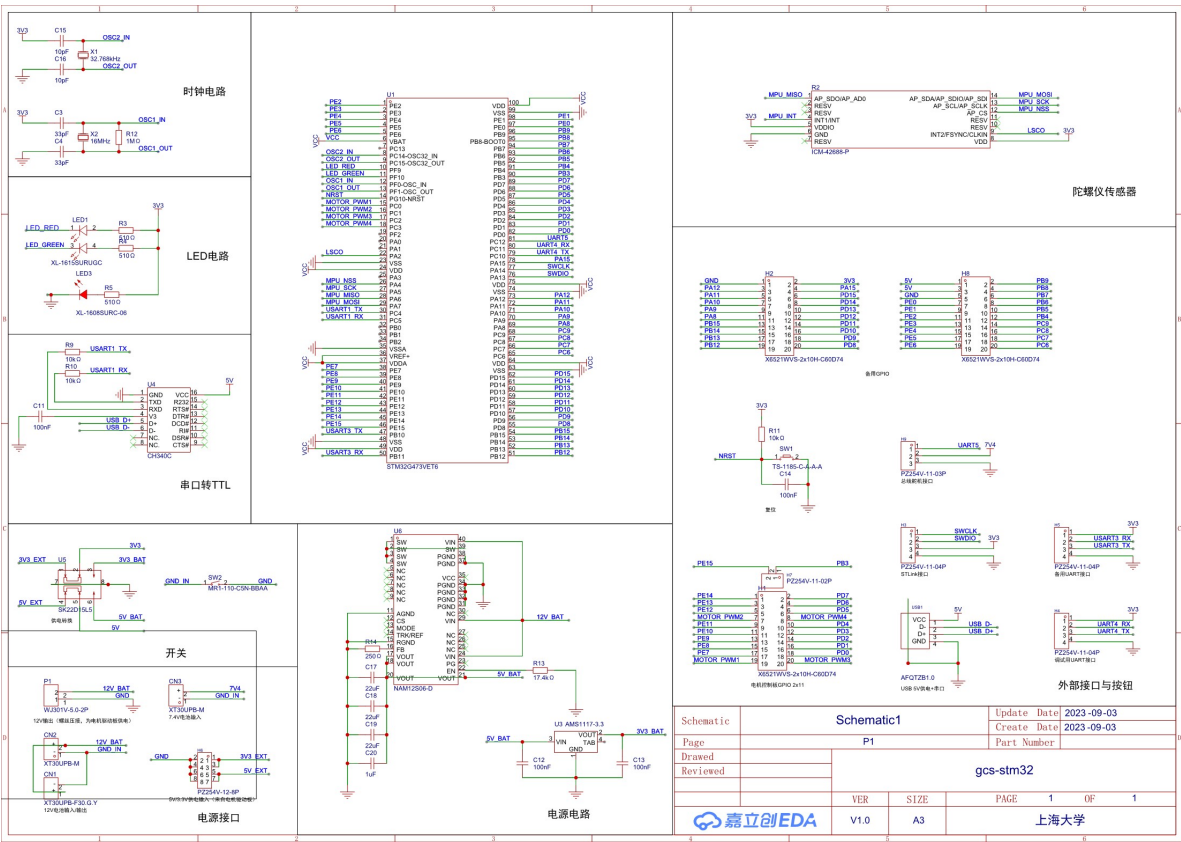
电机和控制方案

电机选择

1. 型号: Golden Motor MagicPie 特点: 适用于电动轮椅、电动自行车、高效、低噪音
2. 小米CyberGear 微电机

这两种 **伺服电机** 都能满足，本系统对于安全性与稳定性的要求，同时能够提供一个较高精度的控制。

控制方案



对于运动主控板我们选择了自我设计一块原创的运动控制模块。主控芯片核心为 **STM32G473VET6**

自主选择元器件，可以让我们在后续的调试和测试中能够有更多的操作空间。

同时由于是自主选择，我们在元器件的选择上，也保留了空余的性能提供给后续的功能拓展。

优势分析：

- 更多的接口：
 - 首先，目前市面流通的板子综合性较强，主控板，空余的接口较少、拓展性差，也会对运行的载负带来影响。而自主设计的主控板针对于赛项，有可以留出更多接口，拓展性强。
- 稳定性：
 - 成品板较为复杂，不易检查故障，而我们设计的主控板更精简，同时我们也更熟悉，可以更快地判断硬件是否除了问题，同时精简的传感器也带来了更好的稳定性。
- 更高精度的传感器：
 - 我们计划使用MPU6500陀螺仪作为传感器，它具有较高的精度，独立设计的主控板允许我们针对任务选择更好的组件。

自动化方案

我们在自动化过程中使用到的传感器有：

- 正前方的高清运动摄像头
- 四周的超声波传感器
- 位于顶部的一个激光雷达
- 内置于主控板上的陀螺仪
- 北斗定位系统

在所有的自动化方案中，智能避障系统的优先度最高且全程开启

智能避障系统

1. 利用前置摄像头传输进入的视频，对于在轮椅前进方向 3米范围内出现人、车、柱子、路肩（俗称马路牙子）、台阶等 障碍物，障碍物的材质包括但不限于透明的、不透明的、反光的、不反光的，落地的 障碍物的高度高于 6cm、宽度大于 5cm 就判定为障碍物，悬空的障碍物的悬空高度低于 90cm、宽度大于 5cm 就判定为障碍物。在轮椅前进方向3米范围内出现深度超过 6cm，宽度超过 8cm 的深坑就判定为 障碍物。在轮椅前进方向3米范围内出现向下的楼梯台阶，判定为重点障碍物。
 - a. 启动激光雷达，对周围环境进行扫描和分析，完成基础地图构建。
 - b. 结合视频信息中的障碍物判断，在建立的地图中进行标注。[重点障碍物附近强行低速通过，同时切入四轮悬挂模式进行下楼梯]
 - c. 将构建好的地图导入由深度学习为基础的模型中进行路线规划。[具体训练方面示例同上]
 - d. 将路线规划数据导入主控板，由主控板控制四轮差速运动进行实际运动。

- e. 在避让过程中使用四周的超声波传感器进行 2 中的判断。
 - f. 将此次的障碍物信息和道路地图存储下来（存储内容为前100m的道路信息）。
2. 利用四周的超声波传感器，检测周围是否有高速移动物体：
- a. 如果有高速移动物体，主控板自动远离高速移动物体。
 - b. 如果有高速移动物体向本体移动，主控板控制本体延对方运动轨迹垂直方向进行规避。[具体方案由AI模型做出决策]
 - c. 在进行规避的同时启动语言提示。
 - d. 完成规避后，自动回归到先前的运动方向。
3. 利用前置摄像头传输进入的视频，若判断前方出现坡道：
- a. 慢速行驶，且通过陀螺仪的速度传感器进行速度限制。

室内路径规划

用户可以在家，通过手机app或者相关可穿戴设备，一键式将轮椅本体驾驶到用户所在处。

1. 通过内置的 星闪 模块接收服务器信息。
- o 补充对于星闪的介绍：目标是推动新一代无线短距通信技术 SparkLink 的创新和产业生态，**该技术可满足智能汽车、工业智造、智慧家庭、个人穿戴等多场景对低时延、高可靠、精同步、多并发的技术需求。星闪 1.0 标准共由 12 项标准组成**，包括接入层、基础服务层、通用（架构、标识等）、安全以及测试等部分。星闪联盟成立于2020年9月，目标是推动新一代无线短距通信技术SparkLink的创新和产业生态，该技术可广泛满足智能汽车、工业智造、智慧家庭、个人穿戴等多场景对低时延、高可靠、精同步、多并发的技术需求。
2. 通过激光雷达和四周的激光传感器对用户家中进行建图并保存。
3. 通过内置的 星闪 模块确定本机和用户位置，通过地图与位置信息放入决策AI模型进行路线设计。
4. 将路线规划数据导入主控板，由主控板控制四轮差速运动进行实际运动。
5. 到达后提示用户上车。

智能导航系统

- 结合线上已有的地图数据与自身的北斗定位系统，与导航厂家合作进行道路规划与智能导航。
- 在实时的运动过程中进行播报。（“转弯”，“直行”，“变向”）
- 利用5G模块，直接调取各大地图导航厂家API，实现精准的导航。

自动化转弯

1. 通过前置的高清运动摄像头对于前方道路是否为弯道进行判断。

```
import cv2
import numpy as np

# 打开视频文件
cap = cv2.VideoCapture('input_video.mp4')

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # 将帧转换为灰度图像
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # 对图像应用高斯模糊以减少噪音
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)

    # 使用Canny边缘检测来检测图像中的边缘
    edges = cv2.Canny(blurred, 50, 150)

    # 对边缘图像应用霍夫直线变换 (Hough Line Transform)
    lines = cv2.HoughLinesP(edges, 1, np.pi / 180, threshold=50,
                             minLineLength=100, maxLineGap=10)

    # 绘制检测到的弯道线
    if lines is not None:
        for line in lines:
            x1, y1, x2, y2 = line[0]
            cv2.line(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)

    # 显示帧, 包含检测到的弯道线
    cv2.imshow('Lane Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# 释放视频捕获对象和关闭窗口
cap.release()
cv2.destroyAllWindows()
```

2. 在判断为正常弯道的情况下进行转弯, 通过车侧边的超声波传感器检测与道路边缘的位置, 软件拟合道路边缘线, 通过主控板来控制四个电机的转速不同来进行差速转弯, 并且使运动轨迹与道路边缘线尽可能的平行。
3. 在运动过程中
4. 在检测到道路重新恢复为直道以后纠正各电机转速, 使整个系统正常执行。

一键掉头或者退出狭窄道路

1. 在用户点击自动转弯以后由四周的超声波传感器检测周围环境是否满足掉头的条件（大于1.45平方米），如果满足则进行：
 - a. 全程进行外放提醒——“掉头请注意”。
 - b. 启动激光雷达，对周围环境进行扫描和分析，完成基础地图构建。
 - c. 将构建好的地图导入由深度学习为基础模型中进行路线规划。[具体训练方面示例同上]
 - d. 将路线规划数据导入主控板，由主控板控制四轮差速运动进行实际运动。
 - e. 再由前置摄像头判断完成掉头再次进入直道以后，停止外放提醒。
2. 如果不满足转弯条件：
 - a. 语言播报提醒用户“不满足掉头条件，进入倒车模式”。
 - b. 全程进行外放提醒——“倒车请注意”。
 - c. 发送命令给主控板，使主控板进行后退的电机控制。
 - d. 在后退过程中读取智能避障系统存储的道路信息进行基本的避障处理。
 - e. 启动激光雷达，对周围环境进行扫描和分析，完成基础地图构建。
 - f. 将构建好的地图导入由深度学习为基础模型中进行路线规划。[具体训练方面示例同上]
 - g. 在后退过程中通过后置的超声波传感器判断后方是否安全。
 - h. 同时使用四周的超声波传感器检测周围环境是否满足掉头的条件（大于1.45平方米），若后续满足则进行1中步骤。

云端存储处理方案

1. 利用内置的北斗定位模块，记录运动轨迹，速度与时间，同时将数据上传云端。信息通过星闪技术和5G通信技术将数据上传。
 - a. 通过星闪协议将数据传输到手机或者手环app中，由app上传到服务器。
 - b. 通过内置的5G通信芯片将信息上传到服务器。
2. 通过手机app或者本地储存，储存7天内的行车视频数据（720P@30fps）
3. 可以将客户针对于系统的设置同步到云端，一键配置新车。
4. 可定制化提示语言，可提供客户的音频资料将训练处模拟客户音色、语气、用于习惯的AI语音。
5. 云服务器使用自动选择算法，将不同地区的数据都传输到最近的数据中心。
6. 与阿里云、腾讯云、华为云合作，共同搭建高效的数据传输途径。

整体安全控制方案

车身行驶安全方案

- 使用现有的成熟框架—— **比亚迪云辇** 系统，四轮独立悬挂，丢轮能跑，稳定性高。
 - 云辇系统的所包含的基本组成单元：
 1. CDC(Continuous Damping Control)电磁悬架
 2. 空气弹簧
 3. 智能液压控控制系统
 4. 相关的视觉传感器，中央控制处理器，执行器
 - 云辇系统运行逻辑：
 1. 视觉传感器（摄像头）判断前方道路情况。
 2. 道路情况汇总到中央控制器和处理器，进行计算后下达指令到执行器。
 3. 执行器作动调节整个底盘系统。
 - 云辇作用列举：
 1. 跑直线时急刹车，纵向的前后俯仰比较明显，云辇通过调节前后悬架高度进而改善车身姿态，增加整车的控制性。
 2. 转向时（特别是转向伴随着大过载的G值时），侧向倾斜幅度比较明显，容易导致车身重心转移，云辇调节一侧的轮胎被顶起，给予一个明显的反向支持。车辆弯道性能得到提升。
- 综上所述，在轮椅上使用 **比亚迪云辇** 系统能够有效提高本机的稳定性与刹车安全性。

使用安全方案

- 内置华为mate60pro同款北斗卫星通讯芯片 **RX6003EQK**，可一键报警与联系紧急联系人。
- 在车身接触皮肤的地方添加血压、血脂、血糖、心率检测，实时监控用户健康状况，突发情况自动求救。
- app内部可以设置定时吃药、定时提醒等功能。车身提供药盒，需要可以自动弹出。

客户端应用控制方案

应用程序框架

1. **模块化架构**：采用模块化设计，以便在不同平台上可以选择性地使用不同的技术栈和框架。

跨平台支持：

1. **选择适当的技术栈：**根据每个平台的需求和特点，选择适用的技术栈。例如，对于桌面平台可以选择Qt，对于移动平台可以选择React Native或Flutter，对于Web平台可以选择React或Angular。
2. **充分利用跨平台工具：**对于那些可以重复使用的模块和组件，充分利用跨平台工具和库，以减少开发工作。

用户界面设计：

1. **不同平台的UI库：**根据不同平台的设计指南，使用不同的用户界面库，以确保应用程序在各个平台上都有最佳的用户体验。

数据同步和存储：

1. **网络通信库：**根据不同平台的网络通信需求，选择适当的网络通信库。例如，对于桌面应用程序，可以使用Qt的网络模块；对于Web应用程序，可以使用Axios或Fetch等。
2. **本地存储：**根据不同平台的需求，选择适当的本地数据存储技术，如SQLite、IndexedDB或Core Data。

用户认证和授权：

1. **OAuth和JWT库：**根据不同平台的需求，选择适当的OAuth和JWT库来实现用户认证和授权。

离线支持：

1. **离线模块：**为不同平台设计并实现离线支持模块，以允许用户在没有网络连接时访问数据。

安全性：

1. **数据加密库：**选择适用于不同平台的数据加密库，确保数据在传输和存储中得到适当的加密保护。

性能优化：

1. **性能分析工具：**使用适用于不同平台的性能分析工具来识别和解决性能瓶颈。

测试和部署：

1. **多平台测试：**进行全面的多平台测试，确保应用程序在各种设备和平台上都能正常工作。
2. **自动化部署：**使用自动化工具和流程来部署应用程序的不同版本和更新。

网页端开发

1. 前端开发：

- 使用HTML、CSS和JavaScript构建网站的前端。
- 使用前端框架（如React、Angular、Vue.js）来创建交互性组件和界面。
- 集成数据可视化库（如D3.js、Chart.js）以展示数据。
- 创建响应式设计，确保网站适应不同屏幕大小和设备。

2. 后端开发:

- 选择后端技术栈, 如Node.js、Python (Django、Flask) 、Ruby on Rails等。
- 创建后端API以处理用户请求和数据存储。
- 连接数据库, 执行数据存储和检索操作。
- 管理用户身份验证和授权。

3. 数据库设计:

- 选择适当的数据库管理系统, 如MySQL、PostgreSQL、MongoDB等。
- 设计数据库架构, 创建表格、视图和索引。
- 确保数据库安全性, 包括数据加密和访问控制。

4. 数据可视化:

- 使用数据可视化工具或库, 将数据转化为可视化图表和图形。
- 创建交互式数据仪表板, 允许用户探索和分析数据。
- 实时数据更新和动画效果可以增加用户体验。

5. 用户认证和安全性:

- 实现用户注册、登录和密码重置功能。
- 确保用户数据和隐私的安全性, 包括数据加密、防跨站脚本攻击 (XSS) 、SQL注入等。

6. 部署和托管:

- 部署网站到Web服务器, 如Apache、Nginx等。
- 使用云托管服务 (如AWS、Azure、Heroku) 进行网站托管。
- 配置域名和SSL证书以提供安全的访问。

7. 测试和优化:

- 进行功能测试, 确保所有功能正常工作。
- 进行性能测试, 优化网站加载速度和响应时间。
- 收集用户反馈, 并根据反馈进行改进和优化。

8. 上线和维护:

- 将网站上线, 向用户提供访问。
- 定期更新和维护网站, 修复漏洞和增加新功能。